VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SPEAKER DIARIZATION

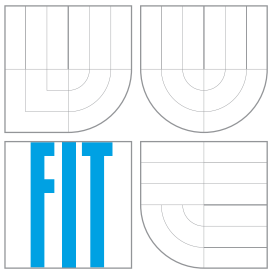
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL TOMÁŠEK

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

KDY KDO MLUVÍ?

SPEAKER DIARIZATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL TOMÁŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL MATĚJKA, Ph.D.

BRNO 2011

Abstrakt

Práce se věnuje implementaci diarizace mluvčího. Popisuje jednotlivé komponenty diarizačního systému, který umí zodpovědět otázku „kdy kdo mluví“. Mezi součásti takového systému patří postupně extrakce příznaků vstupních dat, detekce řeči/ticha, segmentace mluvčích, jejich následné shlukování a nakonec i techniky zaměřené na zlepšení finální segmentace. Práce pochopitelně uvádí i dosažené výsledky implementovaného systému na testovací sadě nahrávek včetně popisu způsobu hodnocení. Testovací nahrávky pochází z NIST RT evaluací z let 2005 – 2007 a nejnižší dosažená chybovost na této sadě je 18,52% DER. K porovnání výsledků systému na testovací sadě souborů je zde uvedena i úspěšnost Marijna Huijbregtse z Nizozemí, který v roce 2009 pracoval se stejnými nahrávkami a dosáhl chybovosti 12,91% DER.

Abstract

This work aims at a task of speaker diarization. The goal is to implement a system which is able to decide “who spoke when”. Particular components of implementation are described. The main parts are feature extraction, voice activity detection, speaker segmentation and clustering and finally also postprocessing. This work also contains results of implemented system on test data including a description of evaluation. The test data comes from the NIST RT Evaluation 2005 – 2007 and the lowest error rate for this dataset is 18.52% DER. Results are compared with diarization system implemented by Marijn Huijbregts from The Netherlands, who worked on the same data in 2009 and reached 12.91% DER.

Klíčová slova

Diarizace mluvčího, segmentace ticho/řeč, segmentace řeči, aglomerativní shlukování, Viterbi algoritmus, statistické modelování směsicí gaussovských rozložení

Keywords

Speaker diarization, voice activity detection, speaker segmentation, agglomerative clustering, Viterbi algorithm, Gaussian mixture modeling

Citace

Pavel Tomášek: Speaker Diarization, diplomová práce, Brno, FIT VUT v Brně, 2011

Speaker Diarization

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Pavla Matějky a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal

.....
Pavel Tomášek
May 12, 2011

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi poskytli odbornou pomoc. A to zejména Pavlu Matějkovi za vedení mé práce, dále i mnohým ostatním členům výzkumné skupiny Speech@FIT, kterým vděčím za četné rady. Rád bych poděkoval všem, kteří mi byli během studia oporou, ale i těm, kteří mi mou cestu jakkoliv znesnadňovali, neboť i díky nim jsem mohl dojít tam, kde jsem.

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2010/2011

Zadání diplomové práce

Řešitel: **Tomášek Pavel, Bc.**
Obor: Počítačová grafika a multimédia
Téma: **Kdy kdo mluví?**
Speaker Diarization
Kategorie: Zpracování řeči a přirozeného jazyka

Pokyny:
Určení kdy kdo mluví v audionahrávce je důležité pro adaptaci rozpoznávače řeči na řečníka a jako předzpracovaní před vlastní identifikací řečníka. Cilem je určit kolik je v nahrávce řečníků, a přiřadit stejné číslo všem segmentům od každého z nich.

1. Seznamte se s algoritmy pro diarizaci řeči.
2. Naimplementujte algoritmus pro diarizaci založeným na Bayesian information criterium (BIC).
3. Ověřte funkčnost na jedné zvukové nahrávce.
4. Seznamte se se strukturou nahrávek v NIST 2005 evaluačních datech.
5. Ověřte funkčnost na NIST 2005 datech.
6. Dosažené výsledky zkodnoťte.

Literatura:

- **David van Leeuwen**: The TNO Speaker Diarization System for NIST RT05s Meeting Data. MLMI 2005: 440-449

Při obhajobě semestrální části diplomového projektu je požadováno:

- 1 - 3

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Matějka Pavel, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 20. září 2010
Datum odevzdání: 25. května 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Contents

List of Figures 4

List of Tables 6

1 Introduction 7
- 1.1 Motivation . . . 7
- 1.2 Structure of Diploma Thesis . . . 7

2 Task of Speaker Diarization 9
- 2.1 Literature Overview . . . 10
- 2.2 Approaches to Speaker Diarization . . . 11
- 2.3 Summary . . . 12

3 Progress 13
- 3.1 Audio Preprocessing . . . 14
- 3.2 Feature Extraction . . . 15
- 3.3 Voice Activity Detection . . . 15
- 3.4 Viterbi Re-segmentation of VAD . . . 17
- 3.5 Speaker Segmentation . . . 20
- 3.6 Agglomerative Clustering . . . 22
- 3.7 Viterbi Re-segmentation of Speaker Clusters . . . 24

4 Experimental Setup 27
- 4.1 Data . . . 27
- 4.2 Evaluation Metrics . . . 27

5 Experiments 30
- 5.1 Specifications of the Initial Configuration . . . 30
- 5.2 System Error Analysis . . . 32
- 5.3 Tuning of Parameters . . . 33
  - 5.3.1 VAD Re-segmentation: Tuning of Number of Gaussians in GMM . . . 33
  - 5.3.2 Speaker Segmentation: Tuning of Number of Training Iterations . . . 34
  - 5.3.3 Re-segmentation of Speaker Clusters: Tuning of Probability of Staying in the Same State . . . 34
  - 5.3.4 Speaker Clustering: Tuning of Lambda . . . 34
- 5.4 Specifications of the Best Configuration . . . 37
- 5.5 Analysis of the Output . . . 39
- 5.6 Comparison with System Implemented by Marijn Huijbregts . . . 40

5.7 Comparison with Other Systems 41
5.8 Speed of the System 42

6 Conclusions and Future Work **44**
6.1 Confrontation with Submission 44
6.2 Utilization of Implemented Speaker Diarization 44
6.3 Future Work 45

References **46**

Glossary **49**

List of Appendices **51**

A Configuration of Feature Extraction **52**

B Detailed Results **53**

C Contents of CD **61**

List of Figures

2.1 Example of input signal (one channel) . . . 9
2.2 Example of final segmentation containing two well detected speakers (wrapped up in orange and green boxes) . . . 9
2.3 Illustration of the two main approaches to clustering (figure from [2, page 25]) 12

3.1 Components of implemented bottom-up architecture . . . 13
3.2 Signal of our experimental recording . . . 14
3.3 Original signal of our experimental recording . . . 16
3.4 Mean normalized energy of our experimental recording . . . 16
3.5 Initial energy-based classification of our experimental recording (speech class is represented by red circles, silent speech and noise are represented by green crosses and silence is represented by blue dots) . . . 18
3.6 Energy-based classification of our experimental recording after 20 iterations (speech class is represented by red circles, silent speech and noise are represented by green crosses and silence is represented by blue dots) . . . 18
3.7 Final speech/non-speech segmentation of our experimental recording (speech segments are bounded by red boxes) . . . 19
3.8 Viterbi re-segmentation of two classes non-speech and speech . . . 19
3.9 Curves of posterior probabilities of speech and non-speech classes based on our experimental recording (speech is represented by red line and silence by green dots) . . . 20
3.10 Illustration of segments A, B and merged segment AB used in ΔBIC estimation. The sliding window is moved by *step* all over the processed speech segment (or from the beginning to the ending of processed signal). . . . 22
3.11 Example of ΔBIC curve of the longest speech segment of our experimental recording. The two green vertical lines emphasize found speaker turns (local maximums, where local means uninterrupted row of positive values). . . . 23
3.12 Curves of posterior probabilities of detected speakers in our experimental recording (two speakers are represented by red and blue lines and silence by green dots) . . . 26

5.1 Tuning of number of Gaussians in GMM used in VAD Re-segmentation, scores represent average voice activity detection error rate . . . 35
5.2 Tuning of number of training iterations used in speaker segmentation . . . 35
5.3 Tuning of probability of staying in the same state used in Re-segmentation 36
5.4 Tuning of lambda parameter used in speaker clustering . . . 36

List of Tables

3.1 An example of a transition probability matrix 20
3.2 Demonstration of a distance matrix of clusters of our experimental recording (rows and columns containing only *-Inf* values: 2, 7 and 9, represent non-speech clusters . . . the distance between speech and non-speech clusters must be the worst) 24
3.3 The content of a transition probability matrix if there are three states and the stay probability is set to 0.9 25

4.1 Test set of recordings 28

5.1 Scores of clustered segmentation of original system and system using reference VAD or reference speaker segmentation to reveal the most faulty subsystem (launched 16th November 2010 without final Viterbi re-segmentation) 32
5.2 The average error rates of parts of implemented system (launched 16th November 2010) 33
5.3 Lambda parameter (used in speaker clustering) with corresponding final DER, number of false alarm and missed speakers 37
5.4 Comparison of scores of diarization system using different configurations (5.1 and 5.4). 39
5.5 Statistics of the final segmentation of NIST_20030925-1517 (output of implemented diarization system using the initial configuration) 40
5.6 Comparison of VAD scores of my system (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 19th February 2009) . 41
5.7 Comparison of scores of my system (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 25th February 2009) 41
5.8 This table presents the average processing time of each subsystem of implemented speaker diarization system without audio preprocessing and feature extraction 43

B.1 Scores of clustered segmentation of original system and system using reference VAD or reference speaker segmentation to reveal the most faulty subsystem (launched 16th November 2010 without final Viterbi re-segmentation) 54
B.2 Comparison of scores of diarization system using different configurations. The first configuration (presented in section 5.1) was launched 28th December 2010 The second configuration (presented in section 5.4) was launched 26th January 2011. long list of differences between *Config. 1* and *Config. 2* is presented in section 5.4. Shortly, the main change is in clustering lambda which was extended from 4.8 to 11. 55

B.3 Detailed DER scores of my system using the best configuration (missed speaker time relative, false alarm time relative, speaker error rime relative and total diarization error rate; launched 26th January 2011) 56
B.4 Comparison of VAD scores of implemented diarization system using the best configuration (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 19th February 2009) 57
B.5 Comparison of scores of my diarization system using best configuration (launched 26th January 2011) and system implemented by Marijn Huijbregts using his best configuration (launched 25th February 2009) 58
B.6 Comparison of scores of my system (launched 26th January 2011) and system implemented by Barra-Chicote et al. [4, page 10, table V] (using mfcc, tdoa+icc) on RT07 meeting set 59
B.7 Comparison of scores of my system (launched 26th January 2011) and system implemented by Luque et al. [14, slide 16] using RT06-07 meeting sets . . . 59
B.8 Comparison of scores of my system (launched 26th January 2011) and LIA-EURECOM RT'09 system implemented by Fredouille et al. [10] which ran on selection of RT meetings 60

Chapter 1

Introduction

When I was choosing a topic for diploma thesis I had three important conditions. It had to be extraordinary, interesting and also useful. While I chose very interesting bachelor work which was in speech processing realm (*Recognition and Search in Skype Calls* [23]) I stayed in this area of fast development and impressive possibilities of application. Finally, I chose speaker diarization. This topic was not easy at the beginning, but, with slow improvements of results, work on speaker diarization system came more and more interesting.

Speaker diarization was also a part of my work of my Socrates ERASMUS internship (*Application of Factor Analysis to Speaker Diarization* [24]) in Laboratoire Informatique d'Avignon (LIA, France, from September 2009 to January 2010). My supervisors were Corinne Fredouille and Driss Matrouf. I have also presented this work at Speaker Odyssey workshop in 2010 [25].

1.1 Motivation

This thesis describes speaker diarization as an useful instrument in speech processing area. Speaker diarization give us an answer on question "Who spoke when?" In which fields can be such an answer appreciable? This information can be very useful for instance in speech recognition systems where it can serve as one way of speaker adaptation to improve the recognition results.

Nowadays, many companies record phone calls. Some of them would like to have also the transcript of these recordings. But these recordings are often summed into one channel containing both sides of conversation. This is mainly done to reduce the size of data for storage purposes. For subsequent speech recognition is better to work with separated speakers (this is a possibility how to reduce the recognition error rate significantly). And this is the right situation for speaker diarization. Speaker diarization can split the summed recording into clusters of segments containing only one speaker.

Also the statistics of speaker turns may be relevant in data mining. Quickly changing short utterances on both sides of a telephone speech can be statistically considered as more interesting.

1.2 Structure of Diploma Thesis

The body of this thesis is divided as follows: chapter 2 (Task of Speaker Diarization), includes an introduction of speaker diarization system (approaches, components and uti-

lization). Chapter 3 (Progress) contains step-by-step progress of implementation. Chapter 4 (Experimental Setup) describes test data and evaluation metrics. Chapter 5 (Experiments) includes interesting experiments with implemented system for speaker diarization.

The last chapter 6 (Conclusions and Future Work), reviews what has been done in this work and mentions the most important questions and possible future work. Then come the references, listing all sources of material, glossary and finally the appendix part.

Chapter 2

Task of Speaker Diarization

As written in motivation part, speaker diarization is a helpful tool which answers question "who spoke when".

Two figures are shown in this section to illustrate the purpose of speaker diarization. The first figure 2.1 shows a signal of a speech. This represents an input for speaker diarization system. The next figure 2.2 presents two speakers in the signal. This is the desired output of a speaker diarization system. There are 2 speakers (two kinds of boxes) in this example. There are several ways to reach such an output. In this thesis I introduce one of the ways – speaker diarization system which consists of speech detection, speaker turn detection and speaker clustering part (described in the next chapter 3).

Figure 2.1: Example of input signal (one channel)

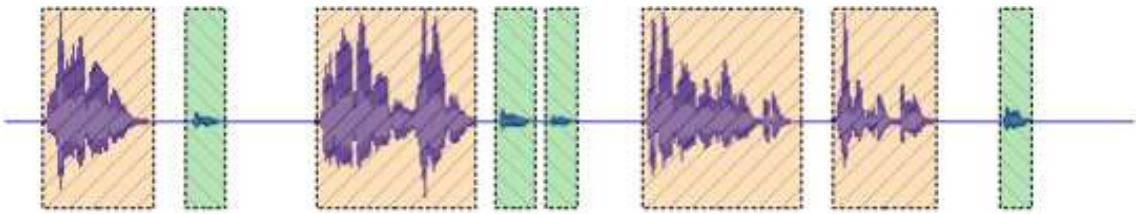


Figure 2.2: Example of final segmentation containing two well detected speakers (wrapped up in orange and green boxes)

2.1 Literature Overview

In this section, bibliography of speaker diarization with brief information about the papers is presented [1]. The first speaker diarization systems were implemented as off-line. Nowadays, most of the systems are still implemented as off-line because it is easier to process a whole recording than an audio stream.

In 2001 Mori and Nakagawa published an article *Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition* [16] where the problems of detection of speaker changes (no information about speakers is available) and clustering algorithm using the Vector Quantization (VQ) distortion measure (Nakagawa and Suzuki, 1993 [17]) are addressed. There is one speaker at the beginning of clustering in the code-book and new speakers whose VQ distortion exceeds a threshold in the current code-book are incrementally added.

In 2005 David van Leeuwen published an article *The TNO Speaker Diarization System for NIST RT05s Meeting Data* [29]. The TNO[2] speaker diarization system is based on a standard BIC[3] segmentation and clustering algorithm and this system is also enhanced by speech activity detector (SAD). The speech detection is based on decoding the speech signal using two GMMs representing silence and speech. The SAD was trained on five AMI meetings data, and tested on other five AMI meetings, performed with a SAD error rate of 5.0%.

The BIC penalty parameter in speaker clustering was optimized to 14. The final speaker diarization error rate on RT05 data was evaluated at 35.1%.

In 2006 Rougui et al. published an article *Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast* [20]. "A GMM based system is proposed, using a modified KL distance between models. Change points are detected as the speech becomes available and data is assigned to either speaker present in the database or a new speaker is created, according to a dynamic threshold. Emphasis is put into fast classification of the speech segments into speakers by using a decision tree structure for speaker models" by [2, page 25].

In 2006 Sylvain Meignier et al. presented an article *Step-by-step and integrated approaches in broadcast news speaker diarization* [15] which summarizes the collaboration of the LIA[4] and CLIPS[5] laboratories on speaker diarization of broadcast news during the spring NIST[6] Rich Transcription 2003 evaluation campaign (NIST-RT'03S). By [15, page 304] "the speaker diarization task consists of segmenting a conversation into homogeneous segments which are then grouped into speaker classes".

In 2006 Xavier Anguera from Universitat Politecnica de Catalunya in Barcelona defended his Ph.D. thesis *Robust Speaker Diarization for meetings* [2]. Presented speaker

[1] Important note: this is only a selection of interesting papers related to speaker diarization. The following short descriptions does not include all the articles ever written about speaker diarization.

[2] *TNO* – Nederlands Instituut voor Toegepaste Geowetenschappen

[3] *BIC* – Bayesian Information Criterion

[4] *LIA* – Laboratoire Informatique d'Avignon, France

[5] *CLIPS* – Communication Langagiere et Interaction Personne-Systeme, Grenoble, France

[6] *NIST* – National Institute of Standards and Technology, USA

diarization system is dedicated to meetings. Also wide background of diarization with details of parts of this system with lots of improvements are mentioned.

In 2008 Marijn Huijbregts from the University of Twente in The Netherlands defended his Ph.D. thesis *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled* [11].

His thesis is about implementation and utilization of diarization, about a system which consists of three subsystems: the speech activity detection subsystem, the speaker diarization subsystem and the automatic speech recognition subsystems. The performance of $SHoUT_{D06}$ system is about 11.76% for twelve meetings of RT05 [11, page 80].

In 2008 Patrick Kenny wrote a technical report about *Bayesian Analysis of Speaker Diarization with Eigenvoice Priors* [13]. Kenny was among others inspired by Variational Bayesian system presented by Fabio Valente [27].

Generally, in Bayesian methods priors are assigned for the parameters [6].

By the words of the author [6, page 14] "work on this system was motivated by the desire to build on the success of factor analysis methods in speaker recognition and to capitalize on some of the advantages a Bayesian approach may bring to the diarization problem (e.g., EM-like[7] convergence guarantees, avoiding premature hard decisions, automatic regularization)".

Kenny used eigenvoice model to represent the speakers. The assumption in eigenvoice modeling is that supervectors (concatenation of the mean vectors in a GMM) have the form [6, pages 14 and 15]:

$$s = m + Vy$$

- s is a randomly chosen speaker dependent supervector
- m is a speaker independent supervector (i.e., UBM)
- V is a rectangular matrix of low rank whose columns are referred to as eigenvoices
- and the vector y has a standard normal distribution; and the entries of y are the speaker factors

In 2009 Douglas Reynolds et al. wrote an article *A Study of New Approaches to Speaker Diarization* [19, page 1047]. This article presents new approaches like "Variational Bayes system using eigenvoice speaker models, a streaming system using a mix of low dimensional speaker factors and classic segmentation and clustering, and a new hybrid system combining the baseline system with a new cosine-distance speaker factor clustering". The best configurations of presented diarization system produced DER from 3.5% to 4.6% on summed-channel telephone speech from the 2008 SRE[8].

2.2 Approaches to Speaker Diarization

There are two main different classes of speaker diarization systems. There are *off-line* systems where is a complete recording to be processed or *on-line* systems where the system can work only with data recorded up to that point.

[7] *EM* – Expectation Maximization
[8] *SRE* – NIST Speaker Recognition Evaluation

Speaker diarization systems can be classified in two main groups by clustering techniques According to Xavier Anguera [2] (2006). Clustering is a method which assigns speech segments into clusters with some kind of similarity (in this case it is the acoustic similarity). These two main approaches are presented in figure 2.3.

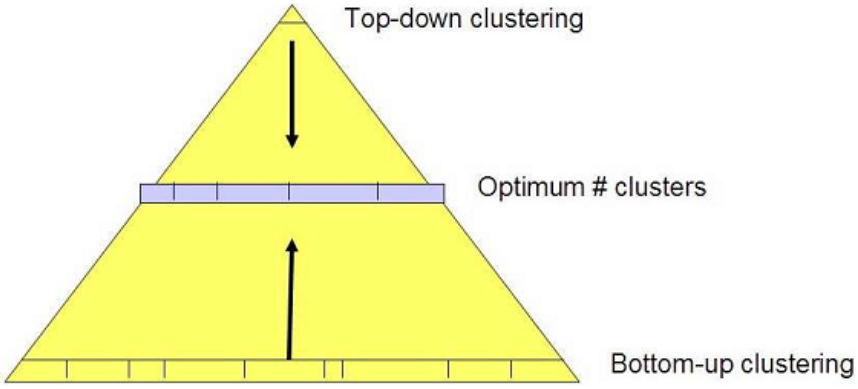


Figure 2.3: Illustration of the two main approaches to clustering (figure from [2, page 25])

Bottom-Up clustering approach (agglomerative) begins with a great number of speech segments. It merges the segments (iteratively computes a distance matrix of clusters and then it merges the closest pair of clusters) until an optimal number of clusters is reached (some kind of criterion and/or other conditions are used there).

Top-Down Clustering Techniques (hierarchical) is a less typical approach and it is a turnover of the previous approach. There is only a few clusters (mainly only one) at the beginning. It iteratively splits/merges the clusters until an optimal number of clusters is reached (some kind of criterion and/or other conditions are used there).

In the literature can be also found combinations of mentioned clustering techniques. But there are not only these two approaches. For example Fabio Valente used a Variational Bayesian learning technique for clustering [27, 28]. It computes the optimum clustering for a range of different number of clusters and uses a distance called *free energy* to determine the optimum. Tsai and Wang [26] worked on a genetic algorithm to obtain an optimum speaker clustering.

2.3 Summary

I decided to implement an **off-line** speaker diarization system using **bottom-up approach**. The main reason for this choice relates to the fact that there were more papers devoted to the first (bottom-up) approach in the time when I started implementing speaker diarization (2008). Therefore I was able to start, implement and experiment with a knowledge from predecessors. For me it was also a good motivation from the psychological point of view because I knew where and how to begin. In the case of second approach I would not have sufficient amount of scientific materials to start my implementation.

Chapter 3

Progress

All components important for bottom-up approach are shown in figure 3.1. Every block in this figure will be described in the sections below.

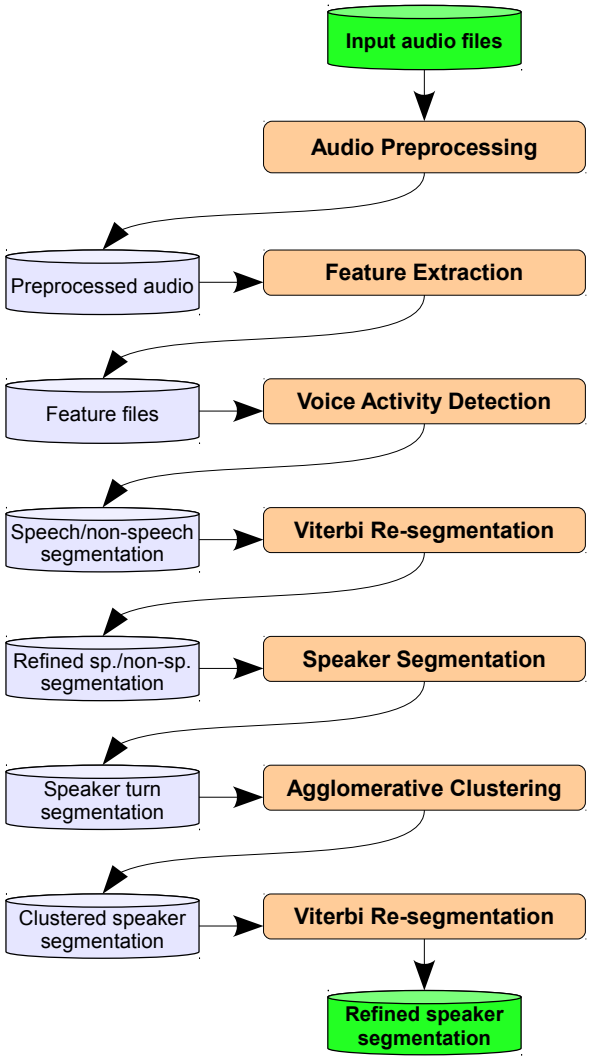


Figure 3.1: Components of implemented bottom-up architecture

Let's have an experimental recording like the one shown in figure 3.2. This signal will serve as a demonstration sample used in the following subsystems for visual illustration. To complete information about this recording I can mention that the duration of this audio file is very short. It has only 22 seconds but it is enough for illustrational purposes. The recording contains two speakers (a woman and a man).

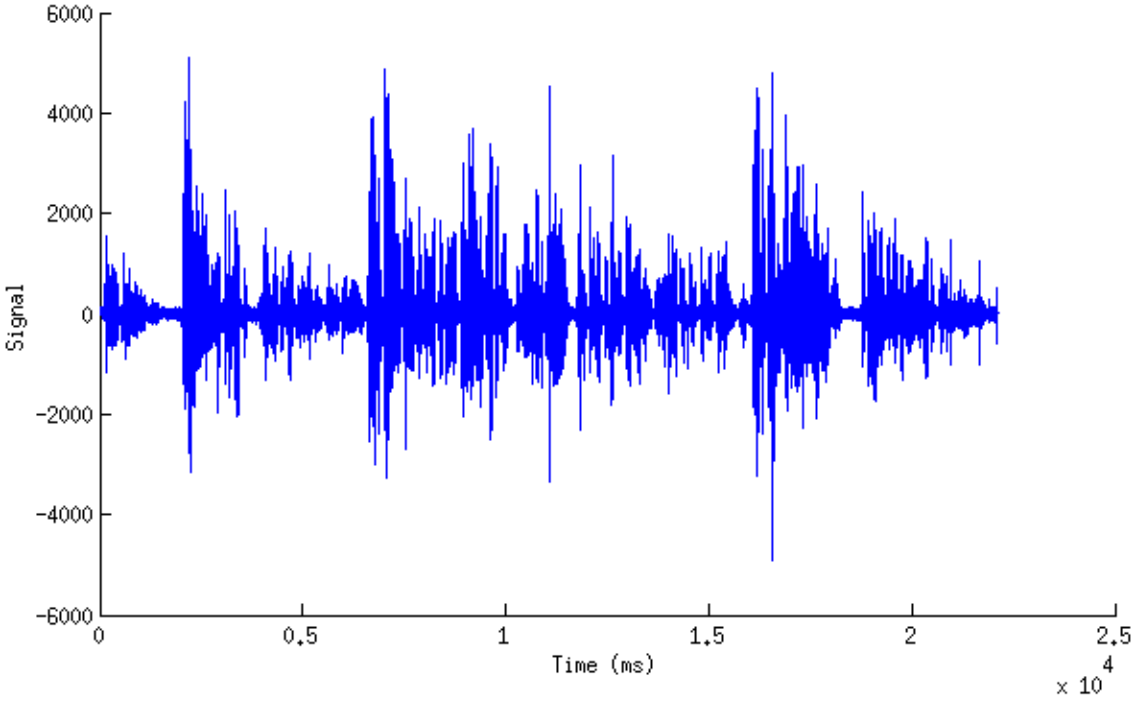


Figure 3.2: Signal of our experimental recording

3.1 Audio Preprocessing

The test data which I was experimenting with (listed in chapter 4) were already preprocessed. The test data came from Marijn Huijbregts who was working with this data in 2009 (and at the end of this thesis I compare results of implemented system with results of his system in the best configurations, see section 5.6). Therefore this preprocessing is not implemented in my diarization system. I mention the following two paragraphs only to inform how the data were modified and to serve a complete information about the history of files I was working with.

The meetings in all RT evaluations (presented in chapter 4) are recorded with multiple distant microphones (MDM) [11]. To gain better results of diarization system the audio signal of each microphone should be first passed through a Wiener filter for noise reduction where noise is assumed to be additive and of a stochastic nature [31]. The implementation of the Wiener filtering was taken from the noise reduction algorithm developed for the Aurora 2 front-end proposed by ICSI, OGI and Qualcomm [1].

After noise reduction, the channels are combined into one "enhanced" channel. They

are not simply summed but a beamforming software is used (BeamformIt[1]). The delay of each signal is computed relatively to the other signals. Before summing all the signals together this delay is removed [2].

3.2 Feature Extraction

Feature extraction is a procedure which produces set of features (feature vectors) which are extracted from original acoustic signal. This is also done to reduce the size of input data which is usually very large and highly redundant

The most commonly used feature extraction methods are MFCC (Mel Frequency Cepstral Coefficients) and PLP (Perceptual Linear Predictive) [11, page 28]. Some experiments show that systems using MFCC can be a little bit better performing than systems with PLP features [2, page 55].

Extracted features are used in the following parts of the system:

- Speaker segmentation (turn detection, section 3.5)
- Agglomerative clustering (section 3.6)
- Re-segmentation using Viterbi algorithm (section 3.4 and 3.7)

For simplification I used only one kind of features in all mentioned parts. I decided to use MFCC consisting of 19 Mel-cepstral coefficients. I used "HCopy"[2] tool To extract the features. Complete configuration file is attached in Appendix part A.

Detailed settings of feature extraction (frame rate, window, ignored sound frequencies and other information) are written in section 5.1.

3.3 Voice Activity Detection

Generally, the aim of voice activity detection is in distinguishing between speech and non-speech (including silence and all kinds of noise). Speech segments in final VAD segmentation should be faultlessly bounded.

I chose VAD based on energy of input signal. No features are used there, just the raw signal which is transformed into frames with length of 20ms (represents 320 samples when working with raw recording in 16000 sampling rate format) and overlap of half length: 10ms.

The energy of frames is computed and mean normalized like shown in figure 3.4. Long recordings are divided into chunks of 30 minutes (to improve memory consumption).

There are three models (one Gaussian per model) which represent speech, noise and silence. These Gaussians are initialized as follows: all weights are set to one third, also covariance matrices are the same for all models (they are computed from the whole chunk), but means differ:

[1] *BeamformIt* is an acoustic beamforming tool developed by Xavier Anguera that accepts a variable amount of input channels and computes an output via a filter&sum beamforming technique [3], available at http://www.xavieranguera.com/beamformit/.

[2] HCopy is a tool from HTK toolkit, by The HTK Book [22, page 81] it is a "general-purpose tool for copying and manipulating speech files").

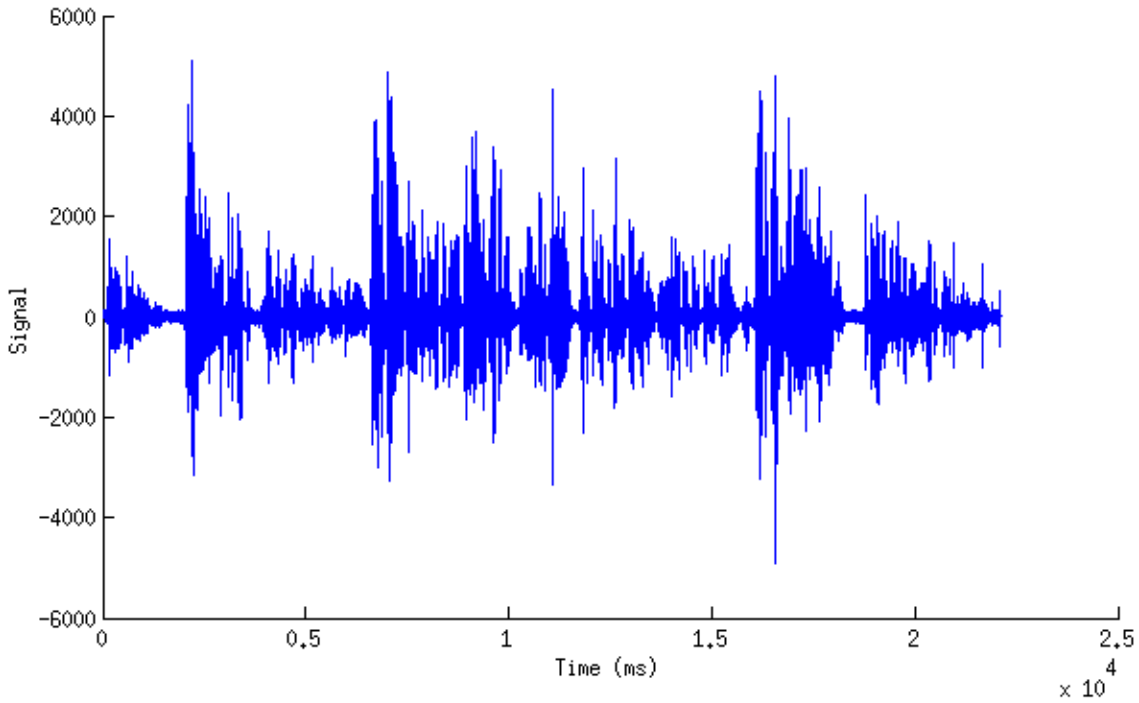


Figure 3.3: Original signal of our experimental recording

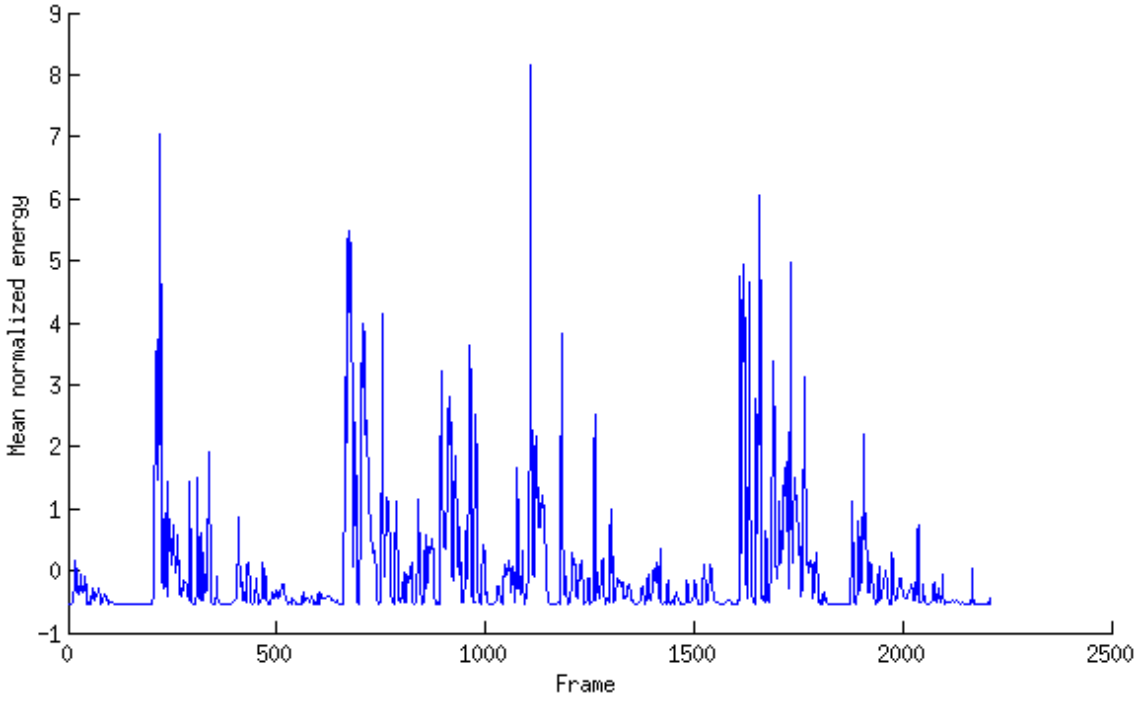


Figure 3.4: Mean normalized energy of our experimental recording

- Mean of the first Gaussian is initialized to the maximal energy (this will represent a speech),
- Mean of the second Gaussian is initialized to the mean energy (this will represent silent a silent speech and a noise)
- And mean of the third Gaussian is initialized to the minimal energy of input signal (this will represent a silence)

Figure 3.5 displays the initial classification of our experimental recording.

Models are iteratively trained using expectation maximization (EM) algorithm until the gain of total log-likelihood is less than a threshold. Figure 3.6 illustrates the classification after 20 iterations.

After this, models are evaluated on all processed chunk. Frames belonging to the first model are assigned to be speech. Frames belonging to the second model are assigned to be speech only if the posterior probability of this Gaussian is higher than a threshold, otherwise they are considered to be some kind of noise (non-speech). And finally, the frames belonging to the third model are assigned to be silence (non-speech).

Postprocessing: After the evaluation speech segments are wrapped up. This means that a certain margin is added on both sides of speech segments. Than, if there is a short gap between two speech segments the gap is removed and speech segments are concatenated. And at the end, very short utterances are eliminated.

Settings of size of margin, gaps to be smoothed and short utterances to be removed of my implementation etc. are specified in section 5.1.

The final speech/non-speech segmentation of our experimental recording is presented in figure 3.7.

3.4 Viterbi Re-segmentation of VAD

Resegmentation of speech/non-speech using Viterbi algorithm tries to iteratively re-train models of speech and non-speech to help the system in decision what is a speech and what is the rest. Several iterations are necessary to improve segmentation.

There are two classes at the beginning of this processing: speech and non-speech. GMM with two Gaussians is prepared for both classes. These models are iteratively trained using expectation maximization (EM) algorithm (using full covariance matrices) until the gain of total log-likelihood is less than a threshold.

Next step uses Viterbi algorithm to estimate which model (two classes: speech/non-speech as in figure 3.8) fits the data best. This is computed for each frame of input signal. The computation is mainly influenced by an acoustic scale coefficient which makes almost sure frames assignments (very high posterior probability for some frames) not so sure and a transition probability matrix (a square matrix containing the probability of staying in the same state and probabilities of transition to other states, these probabilities are set for each state). The transition probability matrix can contain values like the matrix in table 3.1. The numbers in this matrix inform that the probability of staying in the state of *non-speech* is the same as the probability of staying in the state of *speech* an is equal to *0.9*. Then the probability of transition from the state of *non-speech* to the state of *speech* and vice-versa is *0.1*.

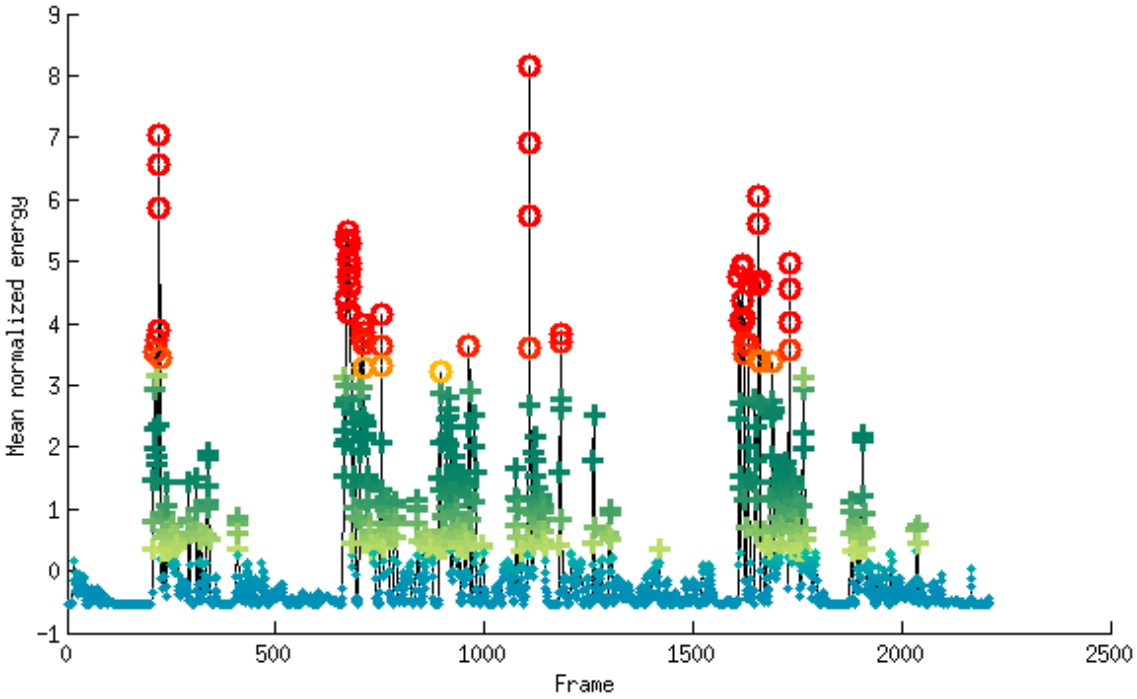


Figure 3.5: Initial energy-based classification of our experimental recording (speech class is represented by red circles, silent speech and noise are represented by green crosses and silence is represented by blue dots)

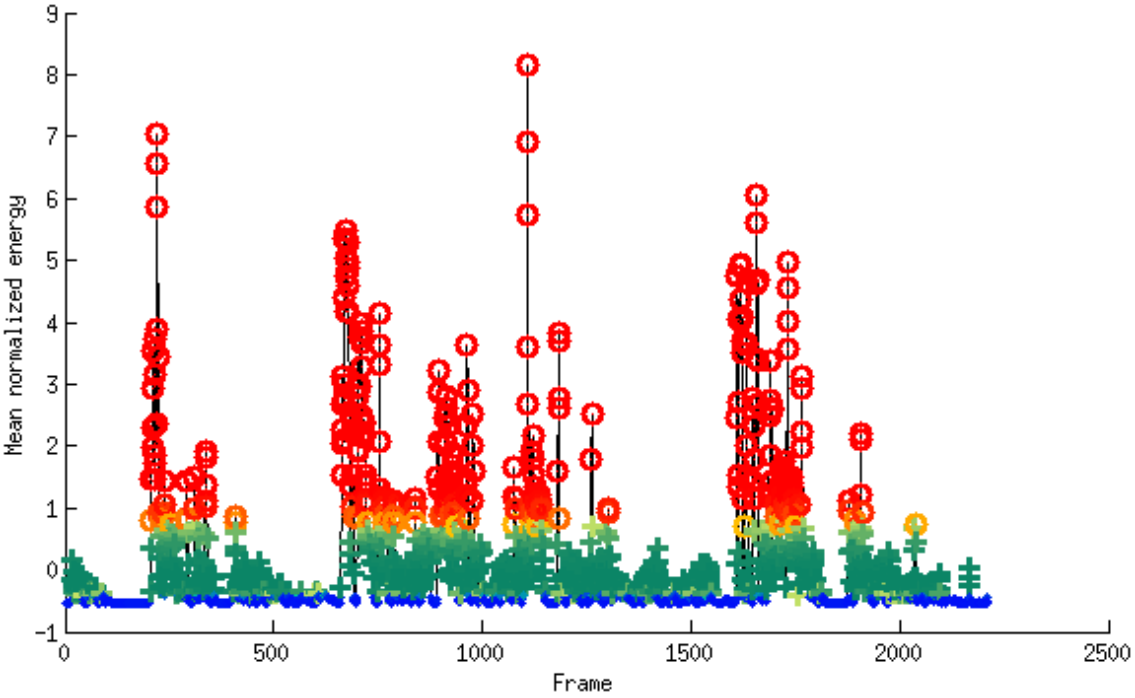


Figure 3.6: Energy-based classification of our experimental recording after 20 iterations (speech class is represented by red circles, silent speech and noise are represented by green crosses and silence is represented by blue dots)

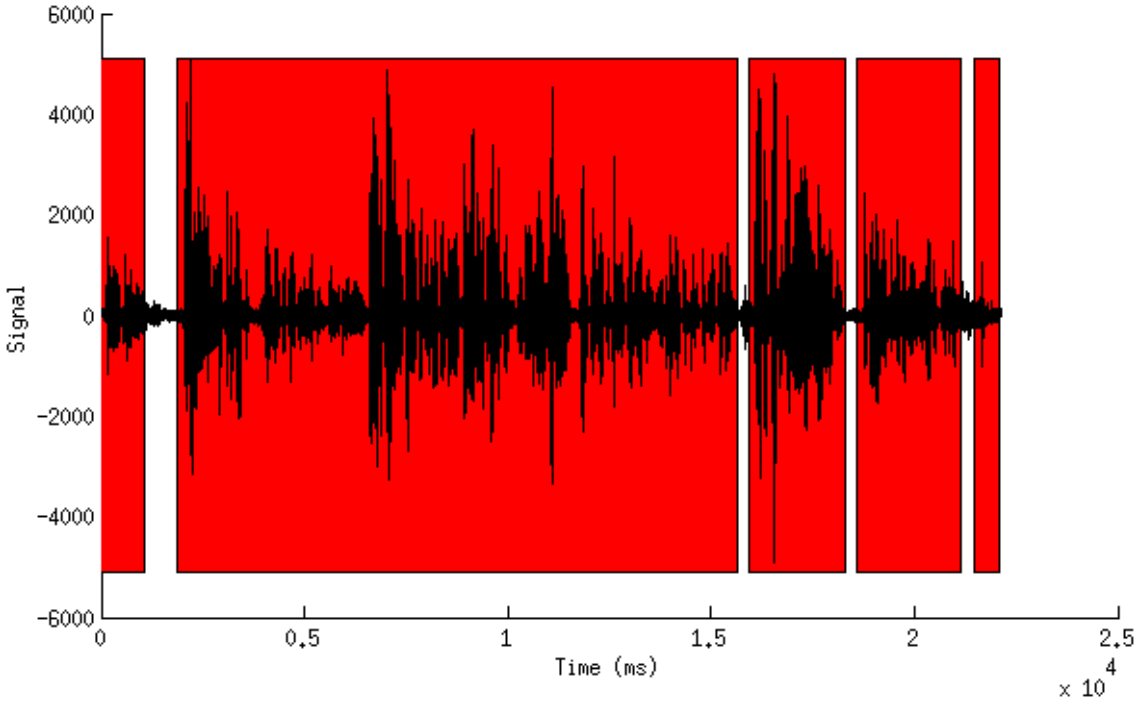


Figure 3.7: Final speech/non-speech segmentation of our experimental recording (speech segments are bounded by red boxes)

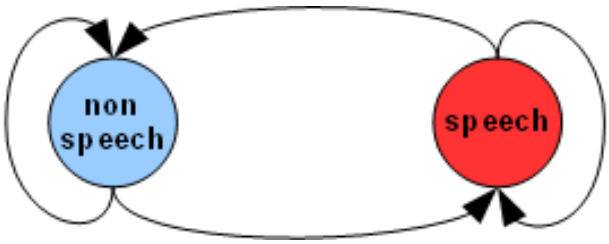


Figure 3.8: Viterbi re-segmentation of two classes non-speech and speech

	Non-speech	*Speech*
Non-speech	0.90	0.10
Speech	0.10	0.90

Table 3.1: An example of a transition probability matrix

The output of implemented algorithm is the best path from posterior probabilities frame-by-frame through two states (speech/non-speech) saying which model fits the data best. The posterior probabilities ("soft decision") are used in the next iteration of re-segmentation process in training of new models.

Postprocessing: Too short utterances are removed from the final segmentation.

Values of coefficients and probability of staying in the same state (from which the transition probability matrix is estimated) used im my implementation are described in section 5.1.

Figure 3.9 displays posterior probabilities of speech and non-speech classes based on our experimental recording at the end of re-segmentation process.

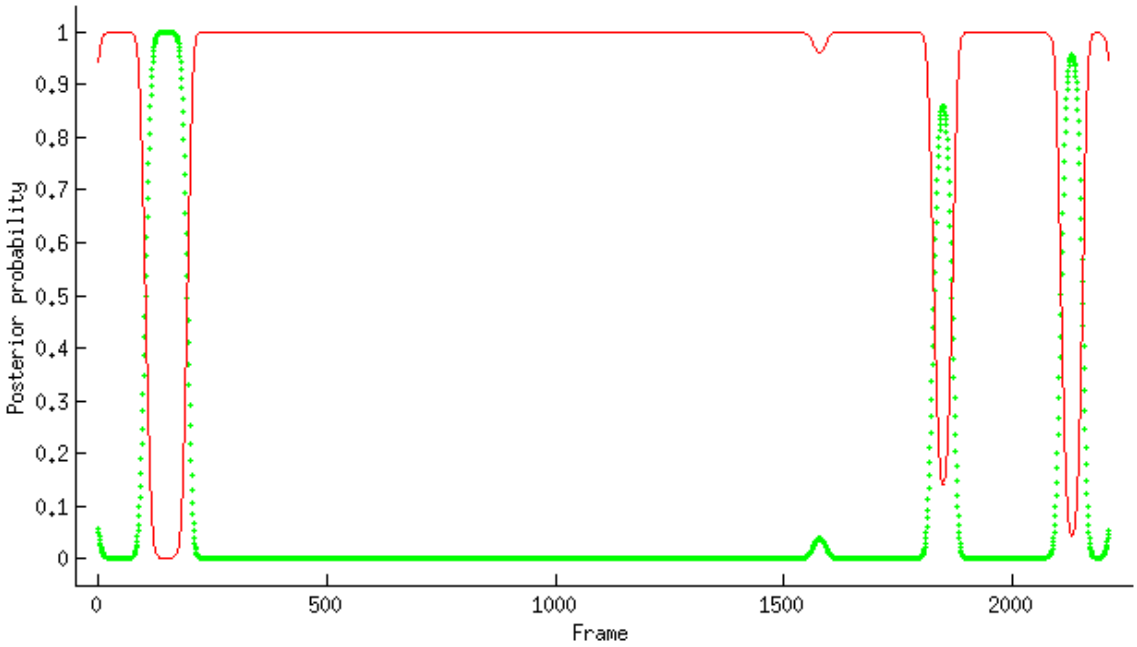


Figure 3.9: Curves of posterior probabilities of speech and non-speech classes based on our experimental recording (speech is represented by red line and silence by green dots)

3.5 Speaker Segmentation

Speaker segmentation tries to find speaker turns in speech segments which are long enough (detection of change points = change of speakers). If such a breakpoint is found speech segment is splitted into two segments with adjusted boundaries. Segmentation containing segments with single speakers only is an output of this process in an ideal case).

Bayesian Information Criterion: The Bayesian Information Criterion (BIC) [21] is the most common distance metric to find the speaker turns. This metric uses model M_i with $\#(M_i)$ parameters representing a segment of data S_i with N_i time frames (feature vectors) and it determines how well the model fits the data [2, 11].

$$BIC(M_i) = logL(S_i, M_i) - \frac{1}{2}\lambda\#(M_i)logN_i \tag{3.5.1}$$

"λ is a free parameter that needs to be tuned on a training set. The value of this parameter influences when the BIC value is positive, meaning that the model fits the data, or negative, meaning that the model does not fit the data very well" [11, page 35].

$\#(M_i)$ is number of free parameters in model (M_i). While computing with full covariance d-dimensional Gaussian distribution (I have not found utilization of diagonal covariance matrix by BIC computation in the literature; full covariance matrix was mentioned in [29, 11, 2]) the number of parameters $\#(M_i)$ representing a segment by [29, page 444] is:

$$\#(M_i) = d + \frac{1}{2}d(d+1) \tag{3.5.2}$$

Formula 3.5.1 can be used to determine if the data of the two segments (simple) S_i and S_j fit M_i and M_j best or if the data of the two segments together (merged) $(S_i + S_j = S)$ fit the model M trained on S the best:

$$\Delta BIC(M_i, M_j) = BIC(M) - [BIC(M_i) + BIC(M_j)] \tag{3.5.3}$$

By [2, 11] formula 3.5.3 can be re-writed into the following formula 3.5.4:

$$\begin{aligned}\Delta BIC(M_i, M_j) = logL(S, M) - [logL(S_i, M_i) + logL(S_j, M_j)] \\ -\lambda\{\#(M) - [\#(M_i) + \#(M_j)]\}logN\end{aligned} \tag{3.5.4}$$

If ΔBIC is negative, the model of the total segment S fits the data not as good as the two separate models and a segment border is placed between the two segments. ΔBIC was first used for segmentation and clustering in [8]. A mathematical proof of formula 3.5.4 is given in [2].

Note that when $\#(M) - [\#(M_i) + \#(M_j)]$ is zero, meaning that the number of free parameters in M equals the number of free parameters in M_i and M_j, parameter λ no longer influences the equation 3.5.4 [11]. Then the final formula is:

$$\Delta BIC(M_i, M_j) = logL(S, M) - [logL(S_i, M_i) + logL(S_j, M_j)] \tag{3.5.5}$$

By [11] the Bayesian Information Criterion has recently been used for speaker change detection and also for speaker clustering in a number of systems [7, 12, 30, 18].

My implementation: The estimation of ΔBIC is exactly according to formula 3.5.5. Full covariance matrices are used when estimating how much does a model fit the data. GMMs modeling simple segments contain only one Gaussian and GMMs modeling merged segments contain twice more Gaussians (two).

Length of a sliding window is set to 2 seconds and BIC step is set to 0.2 second. Detailed settings are in section 5.1.

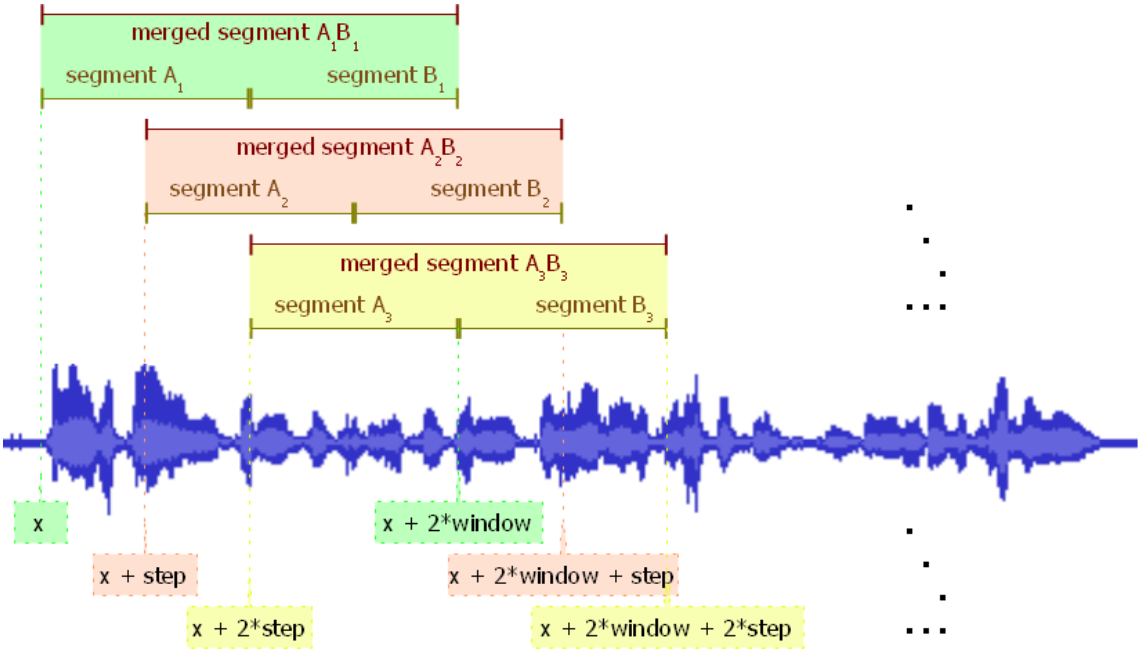


Figure 3.10: Illustration of segments A, B and merged segment AB used in ΔBIC estimation. The sliding window is moved by *step* all over the processed speech segment (or from the beginning to the ending of processed signal).

Figure 3.10 illustrates the processing of a signal. We need segments A, B and merged segment AB to estimate ΔBIC. These segments are moved by *step* from the beginning of a speech segment until the end of the processed speech segment (exactly, until the end of segments B and AB are behind border of processed speech segment). This is done for all the detected speech segments of input signal.

Figure 3.11 shows the ΔBIC curve of the longest speech segment of our experimental recording. The figure is interesting for the fact that it illustrates the way of determination of speaker trun points. Speaker turns are represented by local maximums, where the word *local* means uninterrupted row of positive values (zero is a threshold).

3.6 Agglomerative Clustering

We have many speech segments from the previous step (speaker segmentation 3.5). By iterative computation of distance matrix of all combinations of clusters, the nearest pair of clusters is merged until stop criterion.

The baseline system consists of the following five steps [6, page 14]:

1. Initialize leaf clusters of tree with speech segments.
2. Compute pair-wise distances between each cluster.
3. Merge closest clusters.
4. Update distances of remaining clusters to new cluster.
5. Iterate steps 2-4 until stopping criterion is met.

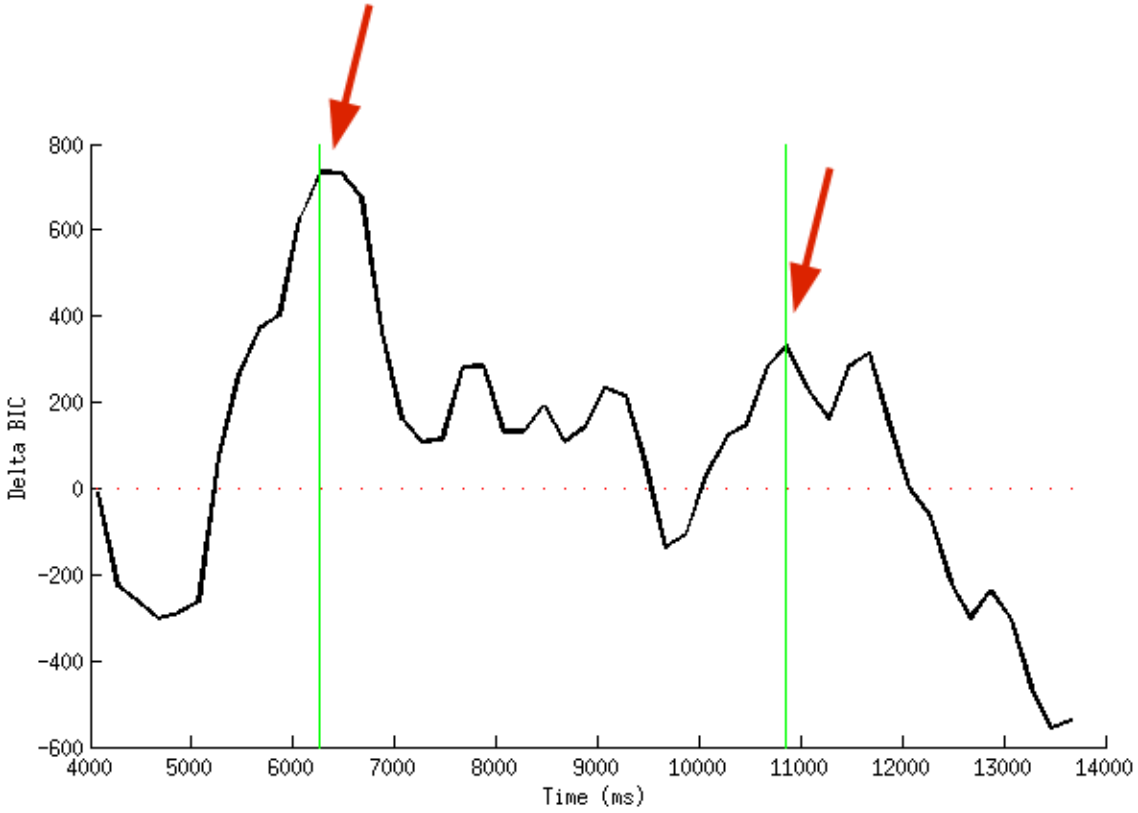


Figure 3.11: Example of ΔBIC curve of the longest speech segment of our experimental recording. The two green vertical lines emphasize found speaker turns (local maximums, where local means uninterrupted row of positive values).

BIC (described in 3.5) serves there as a distance metric and also a stopping criterion. But the tuning parameter λ is needed in clustering phase of my implementation. I was not succesful with removal of λ parameter (everything was merged to one cluster all the time; one of the problems are probably with different amount of assigned frames per cluster). That is why I used original formula 3.5.3 for ΔBIC computation. The final implemented extended formula for clustering is:

$$\begin{aligned} \Delta BIC(M_i, M_j) = logL(S, M) - \frac{1}{2}\lambda \#(M_i) log(N_i + N_j) \\ -logL(S_i, M_i) + \frac{1}{2}\lambda \#(M_i) logN_i \\ -logL(S_j, M_j) + \frac{1}{2}\lambda \#(M_i) logN_j \end{aligned} \tag{3.6.1}$$

In formula 3.6.1:

- M_i and M_j represent models of clusters S_i and S_j
- M represents model of merged cluster $S = S_i + S_j$
- λ is a tuning parameter
- N_i and N_j represent number of frames (feature vectors) of clusters S_i and S_j
- $\#(M_i) = d + \frac{1}{2}d(d+1)$ represents the number of parameters while using full covariance d-dimensional Gaussian distribution [29, page 444]

Table 3.2 contains values of distance (proximity) matrix related to clusters of our experimental recording for demonstration purposes. The highest value is *829.5* in this table. This value is a result of distance test between clusters *1* and *4*. These clusters will be merged (cluster 1 will be enriched by the segments from cluster 4). After merging, distances between the new cluster and the remaining will be re-computed. This process continues until there is no positive number in the distance matrix.

	Cls 1	**Cls 2**	**Cls 3**	**Cls 4**	**Cls 5**	**Cls 6**	**Cls 7**	**Cls 8**	**Cls 9**	**Cls 10**
Cls 1		-Inf	451.3	**829.5**	402.5	697.0	-Inf	673.1	-Inf	519.8
Cls 2	-Inf		-Inf	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf
Cls 3	451.3	-Inf		313.3	784.1	559.5	-Inf	307.7	-Inf	546.7
Cls 4	**829.5**	-Inf	313.3		158.8	656.8	-Inf	600.2	-Inf	483.6
Cls 5	402.5	-Inf	784.1	158.8		429.4	-Inf	87.4	-Inf	440.9
Cls 6	697.0	-Inf	559.5	656.8	429.4		-Inf	728.0	-Inf	553.0
Cls 7	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf		-Inf	-Inf	-Inf
Cls 8	673.1	-Inf	307.7	600.2	87.4	728.0	-Inf		-Inf	527.2
Cls 9	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf		-Inf
Cls 10	519.8	-Inf	546.7	483.6	440.9	553.0	-Inf	527.2	-Inf	

Table 3.2: Demonstration of a distance matrix of clusters of our experimental recording (rows and columns containing only *-Inf* values: 2, 7 and 9, represent non-speech clusters ... the distance between speech and non-speech clusters must be the worst)

Speeding up the clustering: The computation time is too long when working with long recordings (one hour and more). To speed up the clustering I decided to divide input clusters into isolated parts containing less number of clusters with reduced λ. I call this technique *"sub-clustering"*. After processing of all the chunks of clusters all the remaining clusters from all chunks are processed together with original value of λ.

When using appropriate reduction of original λ in sub-clustering it does not lead to worse results. This can speed up the system more than 10 times when processing long recordings.

Postprocessing: Too short utterances are removed from the final segmentation. And if there is a short gap between two segments of a certain speaker (two segments of the same speaker in a row) the gap is removed.

Detailed settings (including λ, reduced λ, number of clusters coming to sub-clustering and other coefficients) are mentioned in section 5.1.

3.7 Viterbi Re-segmentation of Speaker Clusters

Viterbi algorithm is applied to refine segmentation obtained by speaker clustering. This step causes changes in segment boundaries (some segments are moved or reassigned to different clusters, this can be seen as a purification of clusters). Several iterations are necessary to improve segmentation.

Interesting results are obtained when computing also with silence as one of the speakers in the first iteration. The silence is re-trained what causes a refinement of speech/non-speech boundaries.

Clusters (speakers) are represented by GMMs with 4 Gaussians there. These models are iteratively trained using expectation maximization (EM) algorithm (using full covariance matrices) until the gain of total log-likelihood is less than a threshold.

Next step uses Viterbi algorithm to estimate which model fits the data best. The computation is influenced by transition probability matrix (a square matrix where for each state there is a probability of staying in the same state and probabilities of transition to the other states) and acoustic scale coefficient which makes almost sure frames assignments (very high posterior probability for some frames) not so sure.

In my implementation, complete transition probability matrix is estimated by the "probability of staying in the same state". An example: if the stay probability is set to 0.9 and there are three states then the transition probability to the other states is estimated to be 0.05, thus the transition probability matrix contains values like the matrix in table 3.3. The numbers in this matrix inform that the probability of staying in the state of *Speaker 1* is the same as the probability of staying in the state of *Speaker 2* and *Speaker 3* an is equal to *0.9*. Then the probability of transition from the state of *Speaker 1* to the state of *Speaker 2* or *Speaker 3* is *0.05* etc.

	Speaker 1	*Speaker 2*	*Speaker 3*
Speaker 1	0.90	0.05	0.05
Speaker 2	0.05	0.90	0.05
Speaker 3	0.05	0.05	0.90

Table 3.3: The content of a transition probability matrix if there are three states and the stay probability is set to 0.9

After one iteration of Viterbi algorithm we get posterior probabilities of all states (clusters) per frame saying which model fits the data best. These posterior probabilities ("soft decision") are used in the next iteration of re-segmentation process in training of new speaker models.

Postprocessing: The segmentation is smoothed. If there is a short gap between two segments of a certain speaker the gap is removed. In the final segmentation too short utterances are eliminated.

Refined segmentation is illustrated in figure 3.12. This figure displays the curves of posterior probabilities of detected speakers in our experimental recording.

Values of coefficients and probability of staying in the same state used in my implementation are described in section 5.1.

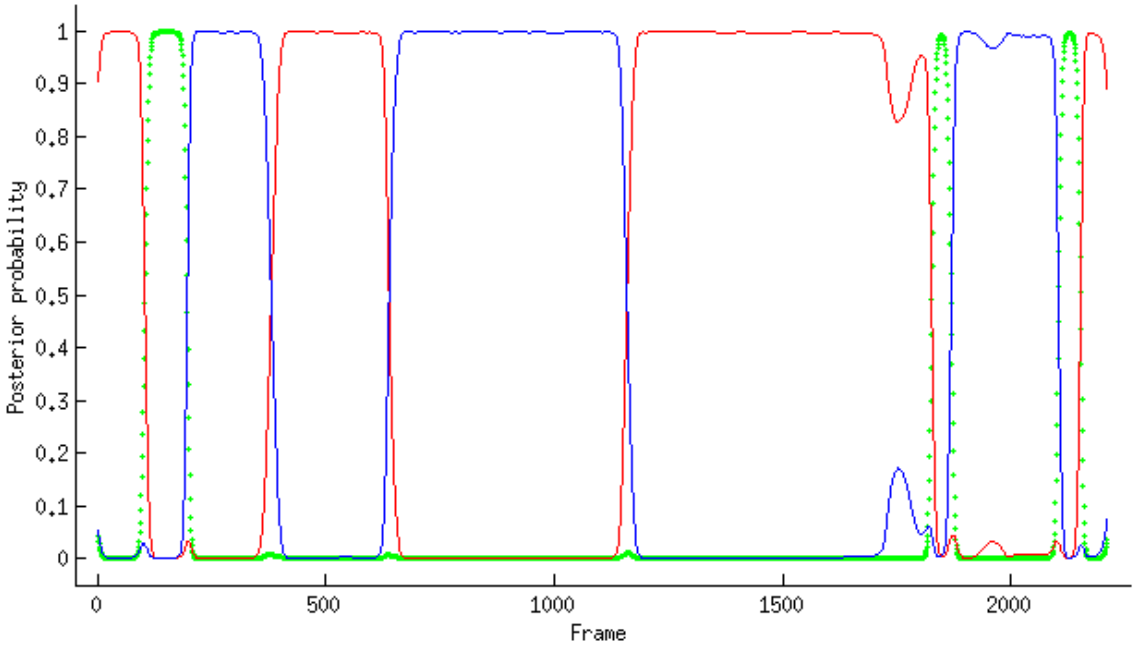


Figure 3.12: Curves of posterior probabilities of detected speakers in our experimental recording (two speakers are represented by red and blue lines and silence by green dots)

Chapter 4

Experimental Setup

This chapter describes test set and a scoring procedure.

4.1 Data

List of used test files is presented in table 4.1 (contains recordings from NIST Rich Transcription Evaluation[1] 2005 – 2007), augmented with durations and numbers of speakers. There is 27 files in the dataset. The total duration is at about 19 hours and 36 minutes. The number of speakers is 128.

There are used only parts of mentioned recordings in scoring by NIST. This limitation is called *unpartitioned evaluation map* (UEM). The total length of scored segments is nearly 7 hours and 38 minutes what is only 38.9% relative from the total duration of all files.

The recordings are in the following format: 16 kHz, mono, 16 bit linear, little endian.

4.2 Evaluation Metrics

To be able to compare results of different diarization systems and its settings we need a tool which analyzes and compares the output of the system to the reference. *Diarization Error Rate* (DER) is a global error rate of diarization system. It consists of:

- Missed speaker time (speech frames are labeled as non-speech frames; so-called MISS)
- False alarm speaker time (non-speech frames are labeled as speech; so-called FA)
- Speaker error time (speech frames of one speaker are labeled as speech frames of another speaker; so-called SPKER)
- Overlap error time (speech frames of two or more speakers speaking in the same time are labeled as speech frames of only one speaker of them and vice versa; so-called OVLER)

In our case we do not count with overlapped speech. Therefore, the complete DER in our case simply relates to MISS + FA + SPKER

For DER estimation I use *"md-eval-v21.pl"* which is a tool from the National Institute of Standards and Technology (NIST)[2].

[1] NIST Rich Transcription Evaluation web page: http://www.itl.nist.gov/iad/mig/tests/rt/

[2] md-eval-v21.pl is available at http://www.itl.nist.gov/iad/mig/tests/rt/2006-spring/code/md-eval-v21.pl

File	Duration (sec)	UEM duration (sec)	# Speakers
AMI_20041210-1052	00:15:44	00:12:10	4
AMI_20050204-1206	00:37:12	00:11:54	4
CMU_20050228-1615	00:18:03	00:12:01	4
CMU_20050301-1415	00:20:08	00:11:58	4
CMU_20050912-0900	00:18:20	00:17:51	4
CMU_20050914-0900	00:19:04	00:17:58	4
CMU_20061115-1030	00:41:17	00:22:29	4
CMU_20061115-1530	00:29:30	00:22:35	4
EDI_20050216-1051	00:29:12	00:18:00	4
EDI_20050218-0900	00:26:56	00:18:10	4
EDI_20061113-1500	00:50:26	00:22:35	4
EDI_20061114-1500	00:48:33	00:22:44	4
ICSI_20000807-1000	01:57:24	00:11:22	6
ICSI_20010208-1430	00:48:27	00:09:59	6
NIST_20030623-1409	00:59:56	00:11:14	5
NIST_20030925-1517	00:40:08	00:11:02	4
NIST_20051024-0930	01:13:28	00:18:08	9
NIST_20051102-1323	01:50:38	00:18:06	8
NIST_20051104-1515	01:10:58	00:22:23	4
NIST_20060216-1347	00:47:24	00:22:28	6
TNO_20041103-1130	00:39:38	00:18:00	4
VT_20050304-1300	00:22:21	00:11:58	5
VT_20050318-1430	00:44:23	00:12:04	5
VT_20050408-1500	00:25:59	00:22:24	5
VT_20050425-1000	00:35:41	00:22:37	4
VT_20050623-1400	00:42:16	00:18:02	5
VT_20051027-1400	00:43:17	00:17:44	4
Total	**19:36:23**	**07:37:56**	**128**

Table 4.1: Test set of recordings

The key part of the computation is the one-to-one mapping of the reference speaker segments to system output. The system does not identify speakers by name and therefore speaker labels can differ from the labels in the reference segmentation [11, page 38].

I used this script with a collar parameter set to 0.25 seconds which represents the no-score boundaries of reference segments. I decided to use this parameter to have the possibility of comparison of results of system implemented by Marijn Huijbregts who used this value in his diarization system in 2009 with this set of files.

By Xavier Anguera [2, page 144] (2006): "When evaluating performance, a collar around every reference speaker turn can be defined which accounts for inexactitudes in the labelling of the data. It was estimated by NIST that a ±250ms collar could account for all these differences.".

I used the evaluation script in this way:

./md-eval-v21.pl -c 0.25 -r ./reference/file.rttm -s ./system/file.rttm -u ./devset.uem

"-c 0.25" is an optional parameter which represents the collar of length of 0.25 seconds. "-r ./reference/file.rttm" specifies the location of the reference segmentation and "-s ./system/file.rttm" specifies the location of the final segmentation of the tested diarization system. "-u ./devset.uem" is an optional parameter which defines a file containing UEM segments for scoring.

Chapter 5

Experiments

This chapter contains specifications of implemented diarization system and selection of interesting experiments.

At first, I started to implement the system in Linux in BASH (shell scripting) using tools like awk, sed, perl and I had to use also STK toolkit[1] and PHNRec[2].

To get deeper in the speaker diarization I changed this approach and I re-implemented the system into Matlab where I had to program complete system (no more STK and PHNRec which I took as "black boxes"). This led me to understanding what is really going on in all of the single parts of speaker diarization. I also got acquainted with technology of GMMs and its training and evaluation. One of the advantages was also the possibility of displaying plots and fast processing of matrices.

5.1 Specifications of the Initial Configuration

The initial specification of implemented speaker diarization system (described in chapter 3) is presented in the following paragraphs. These values were empirically set (all of them except of feature extraction part). These values represent something like a "first shot". They were set during the long process of implementation and they need to be tuned.

Feature Extraction (described in section 3.2)

- Source is "nohead" waveform with rate 625 and "VAX" byteorder (16kHz, little-endian input format, detail of input audio is in 5.5)
- 23 filter-bank channels
- 19 Mel-cepstral coefficients
- 10 ms frame rate, 20 ms window
- Using only frequencies from 64Hz to 8000Hz
- Without preemphasis
- Hamming window on speech frame

[1] *STK toolkit*, developed by Speech@FIT, replaces and enhances HTK toolkit, which is used on Hidden Markov model training, available at http://speech.fit.vutbr.cz/en/software/hmm-toolkit-stk

[2] *PHNRec*, developed by Speech@FIT, is a phoneme recognizer based on long temporal context, available at http://speech.fit.vutbr.cz/en/software/phoneme-recognizer-based-long-temporal-context

- Power spectrum is used
- Features are extracted by HCopy from HTK toolkit, configuration file is attached in Appendix A.

VAD (described in section 3.3)

- 3 models: speech, noise and silence
- One Gaussian per model
- Maximal length of a chunk: 30 minutes
- Minimal improvement in log-likelihood to continue training: 0.00002
- Maximal number of GMM training iterations: 200
- Speech threshold for middle Gaussian: 0.05
- Margin around speech segments (postprocessing): 120 millisecond
- Remove speech segments shorter than (postprocessing): 290 milliseconds
- Minimal gap between speech segments (postprocessing): 300 milliseconds

Re-segmentation of VAD (described in section 3.4)

- 2 GMMs representing speech and non-speech class
- Each GMM is consisting of 2 Gaussians
- Number of refining iterations: 3
- Acoustic scale: 0.05
- Probability of staying in the same state: 0.98
- Minimal gap between speech segments (postprocessing): 200 milliseconds
- Remove speech segments shorter than (postprocessing): 150 milliseconds

Speaker Segmentation (described in section 3.5)

- Simple segments modeled by 1 Gaussian and merged segments modeled by GMMs with 2 Gaussians
- Number of training iterations of model of merged segment: 5
- Length of segment: 2000 milliseconds
- Length of step: 200 milliseconds

Agglomerative Clustering (described in section 3.6)

- GMM represents a cluster
- Each GMM is consisting of only 1 Gaussian
- λ: 4.8
- Sub-clustering (maximal number of clusters to be processed in a chunk): 100
- λ_{sub} for sub-clustering: 0.3λ
- Minimal gap between speech segments (postprocessing): 500 milliseconds
- Remove speech segments shorter than (postprocessing): 350 milliseconds

Re-segmentation of Speaker Clusters (described in section 3.7)

- GMM represents a cluster
- Each GMM is consisting of 2 Gaussians
- Acoustic scale: 0.05
- Probability of staying in the same state: 0.99
- Number of refining iterations: 3
- Minimal gap between speech segments (postprocessing): 100 milliseconds
- Remove speech segments shorter than (postprocessing): 100 milliseconds

5.2 System Error Analysis

This is an analysis of using reference data and the initial configuration (5.1) aimed at revealing the best/worst performing part of the implemented system.

Hypothesis: Using reference segmentation can be helpful in determination of the most faulty part of a system.

Reference VAD segmentation will be used as an input for the speaker segmentation block to reveal the potential of possible improvements made in voice activity detector (the output error depends only on speaker segmentation and clustering). The system will only run speaker segmentation and clustering (without final re-segmentation).

Appropriately adapted reference data will be also used as an input for the speaker clustering block. Scores of this will show how big error can be reduced by improving speaker clustering. The system will only run speaker clustering (without final re-segmentation).

Results: The best and the worst scores of all processed meetings with an average of all meetings from dataset used in this experiment are presented in table 5.1. Detailed scores are presented in Appendix part of this thesis in table B.1.

Original system used in this experiment used VAD, VAD re-segmentation and speaker segmentation with clustering. The second system which was using reference VAD used speaker segmentation with clustering. And finally, the third system which was using reference speaker segmentation data used only speaker clustering. This three system modifications did not use the final Viterbi re-segmentation to be able to see the raw error rate of speaker clustering.

	Diarization error rate (%)		
File	**Original**	**Using ref. VAD**	**Using ref. Speaker Seg**
The best meeting	11.82%	3.14%	1.73%
The worst meeting	65.13%	55.42%	48.35%
Average of all meetings	**33.79**	**24.09**	**20.08**

Table 5.1: Scores of clustered segmentation of original system and system using reference VAD or reference speaker segmentation to reveal the most faulty subsystem (launched 16th November 2010 without final Viterbi re-segmentation)

Analysis: the results presented in table 5.1 show scores of system using reference segmentation to reveal the most faulty subsystem.

The original system produces 33.79% DER and system using reference VAD produces only 24.09% DER in average. The improvement is 9.70% by using reference data. This value represents the average error rate of VAD.

The system which is using reference VAD produces average DER of 24.09%. While using reference segmentation with speaker turns the value of average DER is 20.08%. This means that speaker segmentation (detection of speaker turns) represents very low average error rate of 4.01% in average. 20.08% represents the average error rate of speaker clustering.

The average error rates of parts of implemented system are also written in table 5.2. The worst working subsystem is speaker clustering where is the highest potential of possible improvements.

Subsystem	**Average Error Rate (%)**
VAD	9.70%
Speaker Segmentation	4.01%
Speaker Clustering	20.08%
Complete System	**33.79%**

Table 5.2: The average error rates of parts of implemented system (launched 16th November 2010)

There is a big gap between the lowest and the highest error. In original system the difference is 53.31% ($|65.13\% - 11.82\%|$). When using reference VAD we get similar difference (52.28%) but when using reference speaker turn segmentation the difference is lower (46.62%). This lower gap is affected by using reference speaker turn detection.

Conclusion: The hypothesis is confirmed because we know where is the highest potential of improvement of implemented speaker diarization system. Further work will be aimed at improvement of speaker clustering subsystem.

5.3 Tuning of Parameters

Sets of values must be tested to find the best configuration. Each test is experimenting only with one coefficient at a time. In the following experiments the starting position is the initial configuration (section 5.1). These experiments took a long time, I spent a lot of days with tuning.

All the experiments were done on complete test set of recordings (described in section 5.5). Some experiments are presented below to show the results of testing. Tuning of lambda I consider as the most interesting test (subsection 5.3.4) where the influence of tuned parameter nicely show how the resulting numbers of false alarm and missed speakers are moving.

5.3.1 VAD Re-segmentation: Tuning of Number of Gaussians in GMM

There are only two models in VAD re-segmentation. The first GMM is modeling speech and the second is modeling non-speech. Higher number of Gaussians per GMM can cause overtraining of models and also makes the system slower (the more Gaussians the more

computation). In the initial configuration, the number of Gaussians is set to two. The numbers of Gaussians tested in this experiment are 1, 2, 4, 8 and 16. Results (average voice activity detection error rate after re-segmentation) are presented in figure 5.1.

Conclusion: The best performing system uses GMMs with only two Gaussians where the average voice activity detection error rate after re-segmentation is only 7.61%. These results show that the value in the initial configuration is suitable. Using only one Gaussian models is also good, such a system is simpler and a little faster. The score is only a little bit worse (7.68%). Results of the rest of values are higher than 8% (8.45% for four Gaussians, 8.62% for eight and 10.55% for sixteen Gaussians).

5.3.2 Speaker Segmentation: Tuning of Number of Training Iterations

This number defines a solid number of iterations of model of merged segment (described in 3.5). The numbers of iterations tested in this experiment are 1, 3, 5, 7 and 9 where 5 was the original number of iterations in the initial configuration. Resulting average DERs of system using various values are shown in figure 5.2.

Conclusion: The best performing system uses seven training iterations of model of merged segment. Average DER with this value is 23.93%. The initial configuration with five iterations reached only 24.71% of average DER.

5.3.3 Re-segmentation of Speaker Clusters: Tuning of Probability of Staying in the Same State

The probability of staying in the same state is used in Viterbi algorithm as described in section 3.7. The higher probability of staying in the same state the lower probability of transition. Values near to one prevent from fast transitions between states (fast changing of speakers). The values of probability of staying in the same state tested in this experiment are 0.95, 0.96, 0.97, 0.98, 0.99 and also 0.999. The original probability used in the initial configuration was 0.99.

Conclusion: The best value of probability of staying in the same state is 0.97. We get 19.27% average DER by using this value. This result is better (1.3% gain) than by using the original probability of 0.99.

5.3.4 Speaker Clustering: Tuning of Lambda

Lambda parameter used in speaker clustering has a big influence on the final diarization error rate. This variable was tested on range of values from 1 to 15. Figure 5.4 represents a chart of different values of lambda with corresponding final average DER of complete test set. These values are also presented in table 5.3 augmented with total numbers of false alarm and missed speakers.

Conclusion: We get the best results (below 19% of DER) with lambda set to 10 or 11. When deciding between these two values I chose 11 because of the great balance between number of total false alarm and missed speakers (see table 5.3).

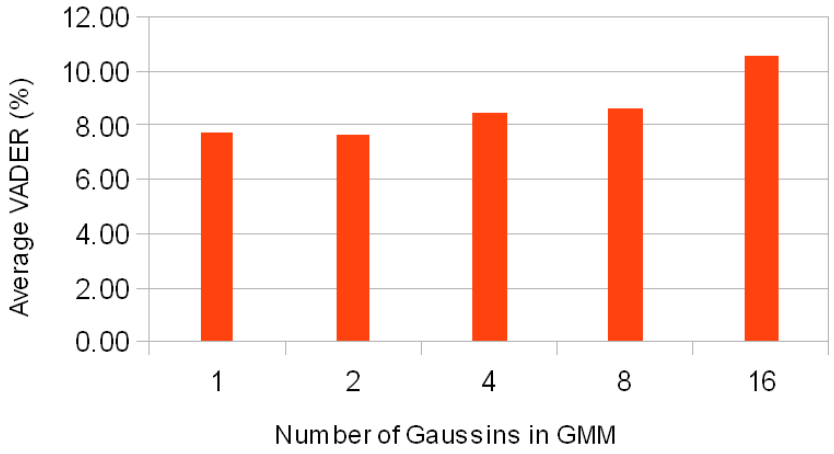


Figure 5.1: Tuning of number of Gaussians in GMM used in VAD Re-segmentation, scores represent average voice activity detection error rate

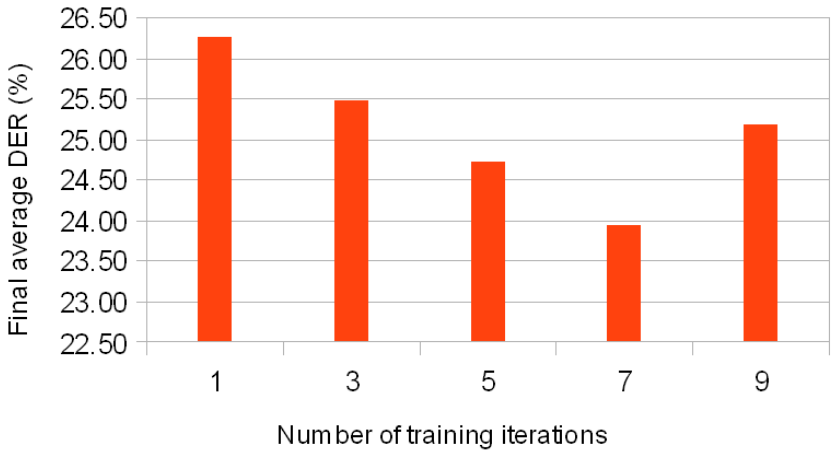


Figure 5.2: Tuning of number of training iterations used in speaker segmentation

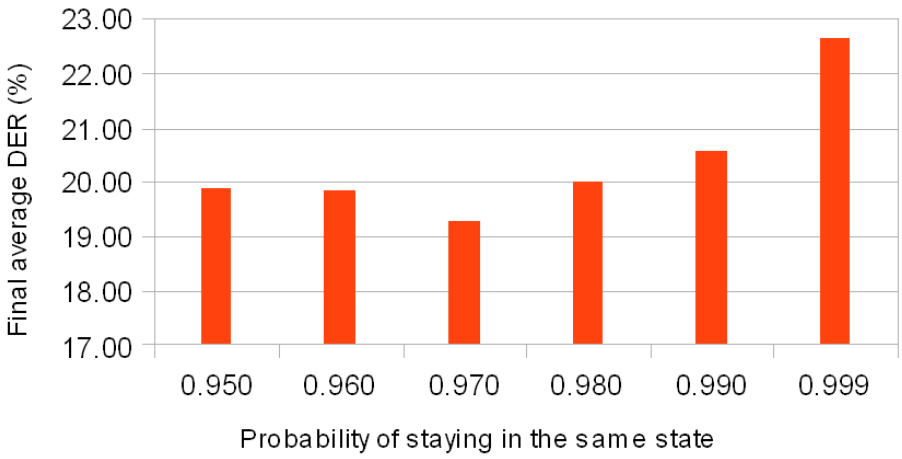


Figure 5.3: Tuning of probability of staying in the same state used in Re-segmentation

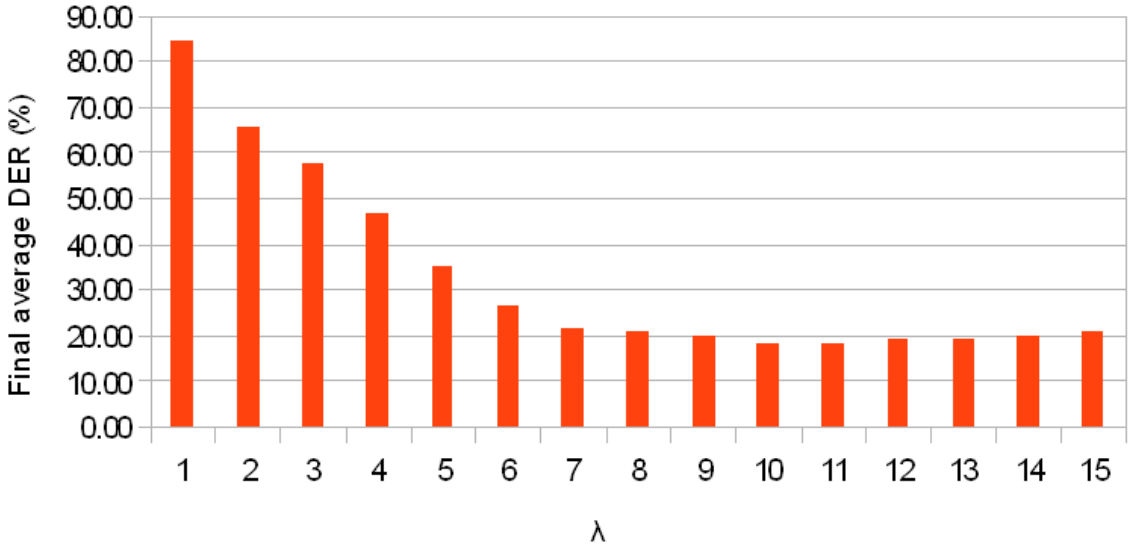


Figure 5.4: Tuning of lambda parameter used in speaker clustering

Lambda	Average DER (%)	# FA Spk	# MISS Spk
1.0	84.34%	18	86
2.0	65.85%	533	0
3.0	57.50%	431	0
4.0	46.53%	264	0
5.0	34.70%	159	0
6.0	26.55%	94	1
7.0	21.64%	65	2
8.0	21.19%	46	5
9.0	19.77%	33	6
10.0	18.44%	19	8
11.0	18.52%	13	13
12.0	19.52%	12	15
13.0	19.38%	8	18
14.0	20.17%	7	24
15.0	21.25%	3	28

Table 5.3: Lambda parameter (used in speaker clustering) with corresponding final DER, number of false alarm and missed speakers

5.4 Specifications of the Best Configuration

Changes of configuration are based on empirical testing of set of values of each variable (some experiments are presented in section 5.3). Modifications were primarily devoted to improve the speaker clustering phase which was analysed as the worst performing part of implemented speaker diarization system (table 5.2 shows the scores using reference data to reveal the worst performing part). Original values of changed variables are mentioned in parenthesis using bold font. The main change is in lambda parameter used in speaker clustering. This tuning parameter was extended from 4.8 to 11 to have less number of clusters at the end of speaker clustering part representing less number of speakers. This results in lower number of false alarm speakers.

Feature Extraction (described in section 3.2) – no changes there

VAD (described in section 3.3)

- 3 models: speech, noise and silence
- One Gaussian per model
- Maximal length of a chunk: 30 minutes
- Minimal improvement in log-likelihood to continue training: 0.05 [**original value: 0.00002**]
- Maximal number of GMM training iterations: 200
- Speech threshold for middle Gaussian: 0.05
- Margin around speech segments: 160 millisecond [**original value: 120 ms**]
- Remove speech segments shorter than: 370 milliseconds [**original value: 290 ms**]
- Minimal gap between speech segments: 150 milliseconds [**original value: 300 ms**]

Re-segmentation of VAD (described in section 3.4)

- 2 GMMs representing speech and non-speech class
- Each GMM is consisting of 2 Gaussians
- Number of refining iterations: 3
- Acoustic scale: 0.05
- Probability of staying in the same state: 0.98
- Minimal gap between speech segments: 0 milliseconds [**original value: 200 ms**]
- Remove speech segments shorter than: 500 milliseconds [**original value: 150 ms**]

Speaker Segmentation (described in section 3.5)

- Simple segments modeled by 1 Gaussian and merged segments modeled by GMMs with 2 Gaussians
- Number of training iterations of model of merged segment: 7 [**original value: 5**]
- Length of segment: 2000 milliseconds
- Length of step: 200 milliseconds

Agglomerative Clustering (described in section 3.6)

- GMM represents a cluster
- Each GMM is consisting of only 1 Gaussian
- λ: 11 [**original value: 4.8**]
- Sub-clustering (maximal number of clusters to be processed in a chunk): 120 [**original value: 100**]
- λ_{sub} for sub-clustering: 0.3λ
- Minimal gap between speech segments: 500 milliseconds
- Remove speech segments shorter than: 400 milliseconds [**original value: 350 ms**]

Re-segmentation of Speaker Clusters (described in section 3.7)

- GMM represents a cluster
- Each GMM is consisting of only 4 Gaussians [**original value: 2**]
- Acoustic scale: 0.05
- Probability of staying in the same state: 0.97 [**original value: 0.99**]
- Number of refining iterations: 5 [**original value: 3**]
- Minimal gap between speech segments: 250 milliseconds [**original value: 100 ms**]
- Remove speech segments shorter than: 250 milliseconds [**original value: 100 ms**]

5.5 Analysis of the Output

Comparison of scores of system using both presented configurations (5.1 and 5.4) are written in table 5.4 which shows the best, the worst and average diarization error rate of all meetings from data set (described in section). Detailed scores augmented with numbers of detected speakers of all meetings are presented in table B.2 in Appendix part of this thesis.

The main information is representing a fact that the new configuration (5.4) successfully reduced average diarization error rate by **5.89%** absolute (24.13% relatively). The biggest gain is due to the higher value of lambda used in speaker clustering which reduced the number of false alarm speakers and makes also a good compromise between missed and false alarmed speakers.

	Diarization error rate (%)	
File	**The initial configuration**	**The best configuration**
The best meeting	VT_20050408-1500: 4.60	VT_20050304-1300: 1.10
The worst meeting	NIST_20030925-1517: 54.81	CMU_20050912-0900: 43.24
Average of all meetings	**24.41**	**18.52**

Table 5.4: Comparison of scores of diarization system using different configurations (5.1 and 5.4).

More statistical information from the output segmentation based on result tables B.2 and B.5 (located in Appendix):

- Number of reference speakers is 128
- Number of speakers detected by my system using the initial configuration is 161 (including 37 false alarm speakers, without 4 missed speakers)
- Number of speakers detected by my system using the best configuration is 128 (including 13 false alarm speakers, without 13 missed speakers), there is a perfect balance between missed and false alarmed speakers
- Number of speakers detected by system implemented by Marijn Huijbregts is 129 (including 16 false alarm speakers, without 15 missed speakers)

One of the reasons why NIST_20030925-1517 using the initial configuration has so high error rate (54.81%) can be seen in the number of detected speakers. Number of real speakers speaking in this recording is only 4 (by the reference data) but number of detected speakers is 7 (see table B.2).

Statistics of the final segmentation of NIST_20030925-1517 using the initial configuration are in table 5.5.

Hypothesis: Looking at table 5.5, if we think only speakers with index 3, 4, 5 and 7 (representing the 4 most speaking speakers, and if we suppose that they are pure – well detected without segments of other speakers) than the residual speakers (indexes 1, 2 and 6) represent 303.2 seconds which corresponds to 12.59% of complete recording duration (silence included). From this point of view we can suppose more than 12.59% gain only by estimating right number of speakers in speaker clustering part of the system.

silence	701.5 sec	(29.1%)
speaker 1	90.0 sec	(3.7%)
speaker 2	174.4 sec	(7.2%)
speaker 3	**404.8 sec**	(16.8%)
speaker 4	**366.6 sec**	(15.2%)
speaker 5	**211.9 sec**	(8.8%)
speaker 6	38.8 sec	(1.6%)
speaker 7	**421.3 sec**	(17.5%)
total	*2408.0 sec*	*(100.0%)*

Table 5.5: Statistics of the final segmentation of NIST_20030925-1517 (output of implemented diarization system using the initial configuration)

Analysis and Conclusion: After running system using the second configuration (the best, section 5.4) we can look at the table comparing scores of the first and the second configuration (B.2). The results show, that our hypothesis for file NIST_20030925-1517 (at least) is confirmed. The number of detected speakers in this file was reduced from 7 to 5. For this file the DER was reduced by **17.84%** (from 54.81% to 36.97%) what is behind predicted limit (12.59% in hypothesis).

Detailed results of implemented diarization system using the best configuration is located in Appendix B.3.

5.6 Comparison with System Implemented by Marijn Huijbregts

This section contains tables which compare scores of my system implemented in Matlab using the best configuration and scores of system implemented by Marijn Huijbregts (also in his best configuration, moreover his system is using delay feature stream[3] [11, pages 89 – 90]).

Presented results of Huijbregts' system were not included in his thesis [11] because they come from improved system based on system described in his thesis. Scores (which are presented below) come from en experiment launched 19th February 2009. I know this from private consultations with Marijn Huijbregts.

VAD Error Rates: Table 5.6 shows the best and the worst voice activity detection error rates. Detailed scores are presented in table B.4 in Appendix part of this thesis.

The average VAD error rate of my voice activity detector is **5.14%** (missed speech: 3.2% and false alarm speech: 2.0%). The average VAD error rate of final output of implemented speaker diarization system (VAD extracted from the final speaker segmentation) of my system is **4.98%** (missed speech: 3.4% and false alarm speech: 1.6%). System implemented by Marijn Huijbregts has **3.27%**. This is a good result for my system. Output scores of my system is worse only by 1.71% absolute.

[3] Utilization of **delay features** can reasonably reduce diarization error rate by 25% relative, "when signals from multiple microphones are available it is possible to use estimates of inter-channel delay as features for diarization." [9]

	VAD error rate (%)	
File	**My system**	**Huijbregts' system**
The best meeting	NIST_20051104-1515: 0.59	NIST_20051104-1515: 0.57
The worst meeting	CMU_20050912-0900: 16.31	CMU_20061115-1030: 7.05
Average of all meetings	**4.98**	**3.27**

Table 5.6: Comparison of VAD scores of my system (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 19th February 2009)

Diarization Error Rates: Table 5.7 shows the best and the worst final diarization error rates. Detailed scores are presented in table B.5 in Appendix.

System implemented by Marijn Huijbregts has **12.91%** DER. The average DER produced by my system was at the absolute beginning (June 2009) at about 50%. After my Erasmus internship in LIA and many school obligations I had time to continue improving my system to reduce this error rate down to 33% (October 2010). In December 2010 I got 24.41%. Finally I reached average DER around **18.52%** (January 2011) what is a very good result for my system. My system is not working with delay features in contrary with the Huijbregts' system. From this point of view the final result is not bad.

	Diarization error rate (%)	
File	**My system**	**Huijbregts' system**
The best meeting	VT_20050304-1300: 1.10	NIST_20030623-1409: 1.73
The worst meeting	CMU_20050912-0900: 43.24	TNO_20041103-1130: 39.71
Average of all meetings	**18.52**	**12.91**

Table 5.7: Comparison of scores of my system (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 25th February 2009)

5.7 Comparison with Other Systems

Not only Marijn Huijbregts works in speaker diarization domain. There are plenty persons and groups all over the World who are also interested in this area of speech processing and have good results. Four of them are mentioned below presenting their diarization error rates.

Here is a comparison of results of my system and system based on intensity channel contribution implemented by Barra-Chicote et al. [4] (presented in 2010). They present a system tested on 8 files of RT07 meeting set with average DER around **13.61%** (using mfcc, tdoa+icc). The average DER of my system on these files is **17.75%**. Detailed scores are presented in table B.6 in Appendix.

A system developed by Bozonnet et al. (presented in 2010) is implementing an integrated top-down/bottom-up approach to speaker diarization [5]. The best DER on RT07 meeting set (the same 8 files as in previous comparison with Barra-Chicote et al.) is **12.9%**. When using only top-down approach the final DER is 15.0% and when using only bottom-up approach the final DER is higher and is equal to 20.8%. If we compare only the bottom-up approaches then the DER of my diarization system is lower by 2.28% absolute.

Next comparison is with UPC RT09 evaluation system by Luque et al. [14] (presented in 2009). Their DER on RT06-07 meeting sets is **23.01%** in average. My system gets only **20.80%** on the same dataset. Detailed scores are presented in table B.7 in Appendix.

Another comparison can be done with LIA-EURECOM RT'09 speaker diarization system [10] (presented in 2009). Its average DER on selection of RT meetings are **14.9%** DER (without scoring overlapping segments). I do not have the same data to be able to to compare the averages. Therefore the detailed scores presented in table B.8 in Appendix does not contain all the results.

Conclusion: Implemented system is not the best, but can surely compete with other speaker diarization systems.

5.8 Speed of the System

By Huijbregts [11] the speed performance of a decoder is often measured with the real-time (RT) factor. The real-time factor is the time that it took to process the test material (P) divided by the duration of this material (L):

$$RT = \frac{D}{L} * 100\%$$

Only the real-time factor is irrelevant. It does not provide sufficient information. The value is hardware-dependent. Therefore we need to know the type of processor and size of memory at least.

Time speed of my diarization system implemented in Matlab was measured on a computer using a Linux system consisting of the following hardware:

- Intel® Core™ 2 CPU 6600 @ 2.40GHz
- 4MB cache
- 1GB main memory

The average speed of implemented speaker diarization system is less than **1/2 real-time** (43/100 exactly). (the processing time is in average twice shorter than the original duration of processed file).

The fastest part of the system is speaker segmentation (described in section 3.5) which needs only 2/100 of real-time in average. Then, the slowest part is the last one: Viterbi re-segmentation of speaker clusters (described in section 3.7). For diarization system configured to run 5 iterations of this re-segmentation, the processing time is about 22/100 of real-time.

To speed up the system with only a little decrease of performance (a little higher DER), we can set only 2 iterations of Viterbi re-segmentation of speaker clusters. Then the processing time of this part would be reduced to 9/100 RT and subsequently the speed of complete diarization system will be only 1/3 RT (34/100). The reduction of number of iterations from 5 to 2 represents only **0.07%** downgrade of final DER (from 18.52% to 18.59%) in implemented diarization system using the best configuration.

Details of each part of the system is described in table 5.8.

It is important to mention, that implemented system was not fully optimized for speed. Main effort was devoted to correct implementation of parts of speaker diarization system

Subsystem	**Speed**
Voice Activity Detection	6% RT
Viterbi Re-segmentation	5% RT
Speaker Segmentation	2% RT
Agglomerative Clustering	8% RT
Viterbi Re-segmentation	22% RT
Total	**43% RT**

Table 5.8: This table presents the average processing time of each subsystem of implemented speaker diarization system without audio preprocessing and feature extraction

enriched by some improvements to reduce the VAD error rate and the final diarization error rate. The system should be completely re-implemented in C or C++ programming language and also the code should be optimized to make it really fast.

Chapter 6

Conclusions and Future Work

This chapter summarises work on speaker diarization system – system which tries to answer "Who spoke when?" It is also including a personal view of the process of implementation and gained experiences. Possibilities of future development are also mentioned.

From personal point of view it was a long way from the first experiments with small system in shell scripts to the fully implemented diarization system with good results. At the beginning it was not easy to find my way but hard work is behind and the system is now working.

This thesis was a good opportunity to get closer to BASH, Perl and Matlab. Programming in mentioned environments/languages led me to reveal a lot of interesting properties of scripting in BASH, Perl and Matlab. I have also extended my knowledge in voice activity detection, Viterbi algorithm, Bayesian information criterion, Gaussian mixture models and speech technologies in general.

6.1 Confrontation with Submission

I got acquainted with basic components of speaker diarization. At the moment of writing documentation for defence of term project, the system was already completely implemented in Matlab. The functionality was tested on a set of files (NIST RT 2005 – 2007) described in chapter 4 (Experimental Setup) and results are presented and analysed in chapter 5 (Experiments).

All the requirements of submission were not only fulfilled but implementation is improved and enriched by many additional advance techniques. The final scores of implemented diarization system are not the best but comparable with other speaker diarization systems.

6.2 Utilization of Implemented Speaker Diarization

As written in introduction, the system can be now utilized as a speaker adaptation technique in speech recognition systems of Speech@FIT group in Brno. Speaker diarization can have a significant influence on overall performance of speech recognition system.

6.3 Future Work

The future of speaker diarization can be seen for example in a relation with factor analysis which can improve performance significantly. The first experiments were already done (see my presentation at Speaker and Language Recognition Workshop, Speaker Odyssey 2010 [25]).

As possible improvements of the diarization system can be considered re-implementation to another programming language (for example C or C++). Also speeding up the system would be very appreciable (it is about 1/2 real-time by now).

Removal of lambda parameter is also a one of the goals for the future. Such a modification would make the system more robust. This parameter is used in speaker clustering and it needs to be tuned for different data (influences merging of clusters).

It is still difficult to cope with speaker overlaps. Implemented system mainly assigns such a segment to a speaker with higher weight (or sometimes simply louder speaker) or the system can create a new speaker as a mixture of the speakers speaking in the same time. Improved system could use for example information about the delay to microphones when processing MDM meetings. Therefore this is also a topic for the future of speaker diarization.

References

[1] Andre Adami, Lukáš Burget, Stephane Dupont, Hari Garudadri, František Grézl, Hynek Heřmanský, Pratibha Jain, Sachin Kajarekar, Nelson Morgan, and Sunil Sivadas. QUALCOMM-ICSI-OGI Features for ASR. In *7th International Conference on Speech and Language Processing*, pages 4–7. International Speech Communication Association, 2002.

[2] Xavier Anguera. *Robust Speaker Diarization for meetings.* PhD thesis, Universitat Politecnica de Catalunya, Barcelona, 2006.

[3] Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic beamforming for speaker diarization of meetings. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 15, issue 7, pages 2011–2023. IEEE, September 2007.

[4] Roberto Barra-Chicote, Jose Manuel Pardo, Javier Ferreiros, and Juan Manuel Montero. Speaker Diarization based on Intensity Channel Contribution. *IEEE Transactions on Audio, Speech and Language Processing*, 2010. pending for publication.

[5] Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Dong Wang, and Raphaël Troncy. An Integrated Top-Down/Bottom-Up Approach To Speaker Diarization. In *Interspeech 2010*, pages 2646–2649. Springer Verlag, Berlin, Germany, September 2010.

[6] Lukáš Burget, Niko Brümmer, Douglas Reynolds, Patrick Kenny, Jason Pelecanos, Robbie Vogt, Fabio Castaldo, Najim Dehak, Reda Dehak, Ondřej Glembek, Zahi N. Karam, John Noecker Jr., Elly (Hye Young) Na, Ciprian Constantin Costin, Valiantsina Hubeika, Sachin Kajarekar, Nicolas Scheffer, and Jan Černocký. Robust Speaker Recognition Over Varying Channels. Technical report, Johns Hopkins University, Baltimore, MD, USA, JHU workshop 2008.

[7] Steve Cassidy. The macquarie speaker diarisation system for RT04s. In *Proceedings of the NIST RT04s Evaluation Workshop*, May 2004.

[8] Scott Shaobing Chen and P. S. Gopalakrishnan. Speaker, Environment And Channel Change Detection And Clustering Via The Bayesian Information Criterion. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop 1997*, pages 127–132, February 1998.

[9] Nicholas Evans, Corinne Fredouille, and Jean-François Bonastre. Speaker diarization using unsupervised discriminant analysis of inter-channel delay features. In *ICASSP 2009, International conference on Acoustics, Speech and Signal Processing*, pages 4061–4064. IEEE, April 2009.

[10] Corinne Fredouille, Simon Bozonnet, and Nicholas Evans. The LIA-EURECOM RT'09 Speaker Diarization System. In *RT'09, NIST Rich Transcription Workshop*, May 2009.

[11] Marijn Huijbregts. *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*. PhD thesis, University of Twente, The Netherlands, 2008.

[12] Dan Istrate, Corinne Fredouille, Sylvain Meignier, Laurent Besacier, and Jean François Bonastre. NIST RT'05S Evaluation: Pre-Processing Techniques and Speaker Diarization on Multiple Microphone Meetings. In *Machine Learning for Multimodal Interaction, Second International Workshop (MLMI'05)*, volume 3869/2006 of *Lecture Notes in Computer Science*, pages 428–439. Springer Verlag, Berlin, Germany, February 2006.

[13] Patrick Kenny. Bayesian Analysis of Speaker Diarization with Eigenvoice Priors. Technical report, Centre de recherche informatique de Montréal, Montreal, Québec, Canada, May 2008.

[14] Jordi Luque and Javier Hernando. Speaker Diarization for Conference Room: The UPC RT09 Evaluation System. In *The Rich Transcription 2009 Meeting Recognition Evaluation Workshop*, May 2009.

[15] Sylvain Meignier, Daniel Moraru, Corinne Fredouille, Jean-François Bonastre, and Laurent Besacier. Step-by-step and integrated approaches in broadcast news speaker diarization. In *Odyssey 2004: The Speaker and Language Recognition Workshop*, volume 20, number 2-3, pages 303–330, 2006.

[16] Kazumasa Mori and Seiichi Nakagawa. Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing 2001*, volume 1, pages 413–416. IEEE, 2002.

[17] Seiichi Nakagawa and Hideyuki Suzuki. A new speech recognition method based on VQ-distortion and hmm. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 676–679. IEEE, 1993.

[18] Elias Rentzeperis, Andreas Stergiou, Christos Boukis, Aristodemos Pnevmatikakis, and Lazaros C. Polymenakos. The 2006 Athens Information Technology Speech Activity Detection and Speaker Diarization Systems. In *Machine Learning for Multimodal Interaction, Third International Workshop (MLMI'06)*, volume 4299/2006 of *Lecture Notes in Computer Science*, pages 385–395. Springer Verlag, Berlin, Germany, May 2006.

[19] Douglas Reynolds, Patrick Kenny, and Fabio Castaldo. A Study of New Approaches to Speaker Diarization. In *Interspeech 2009*, September 2009.

[20] Jamal Eddine Rougui, Mohammed Rziza, Driss Aboutajdine, Marc Gelgon, and Jose Martinez. Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5. IEEE, 2006.

[21] Gideon Schwarz. Estimating the Dimension of a Model. In *The Annals of Statistics*, volume 6, number 2, pages 461–464. Institute of Mathematical Statistics, March 1978.

[22] Young Steve, Evermann Gunnar, Gales Mark, Hain Thomas, Kershaw Dan, Moore Gareth, Odell Julian, Ollason Dave, Povey Dan, Valtchev Valtcho, and Woodland Phil. *The HTK book (for HTK Version 3.3)*. Entropics Cambridge Research Lab., 2005.

[23] Pavel Tomášek. Recognition and Search in Skype Calls. Bachelor's thesis, Brno University of Technology, Faculty of Information Technology, June 2008.

[24] Pavel Tomášek. Application of Factor Analysis to Speaker Diarization. Technical report, Laboratoire Informatique d'Avignon, January 2010.

[25] Pavel Tomášek, Corinne Fredouille, and Driss Matrouf. Factor analysis-based approaches applied to the speaker diarization task of meetings: a preliminary study. In *Proceedings of the Speaker and Language Recognition Workshop, Speaker Odyssey 2010*, 2010.

[26] Wei-Ho Tsai and Hsin-Min Wang. On maximizing the within-cluster homogeneity of speaker voice characteristics for speech utterance clustering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2006.

[27] Fabio Valente. *Variational Bayesian methods for audio indexing*. PhD thesis, Universite de Nice-Sophia Antipolis, 2005.

[28] Fabio Valente. Infinite models for speaker clustering. In *International Conference on Speech and Language Processing*, number Idiap-RR-19-2006 in Lecture Notes in Computer Science, 2006. Published in ICLSP 2006.

[29] David van Leeuwen. The TNO Speaker Diarization System for NIST RT05s Meeting Data. In *Machine Learning for Multimodal Interaction, Second International Workshop (MLMI'05)*, volume 3869/2006 of *Lecture Notes in Computer Science*, pages 440–449. Springer Verlag, Berlin, Germany, February 2006.

[30] David van Leeuwen and Matěj Konečný. Progress in the AMIDA speaker diarization system for meeting data. In *Multimodal Technologies for Perception of Humans 2007*, volume 4625/2008 of *Lecture Notes in Computer Science*, pages 475–483. Springer Verlag, Berlin, Germany, 2008.

[31] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, August 1949. ISBN 02-622-3002-X.

Glossary

AMI	Augmented Multi-party Interaction
AMIDA	Augmented Multi-party Interaction with Distant Access
BIC	Bayesian Information Criterion
CLIPS	Communication Langagiere et Interaction Personne-Systeme (Grenoble, France)
CMU	Carnegie Mellon University (Washington, USA)
DARPA	Defense Advanced Research Projects Agency
EDI	Electronic Data Interchange
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HMM	Hidden Markov Models
HTK	HMM Toolkit
ICSI	International Computer Science Institute (Berkeley, California, USA)
ICSLP	International Conference on Speech and Language Processing
JHU	Johns Hopkins University (Baltimore, Maryland, USA)
LIA	Laboratoire Informatique d'Avignon (France)
MAP	Maximum A Posteriori
MDM	Multiple Distant Microphones
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLMI	Machine Learning for Multimodal Interaction
ms	Millisecond

NIST	National Institute of Standards and Technology (Gaithersburg, Maryland, USA)
OGI	Oregon Graduate Institute (Beaverton, Oregon, USA)
PLP	Perceptual Linear Predictive analysis of speech
RT	Real-Time
	Rich Transcription Evaluation (by NIST)
SAD	Speech Activity Detection
spk	Speaker
SRE	Speaker Recognition Evaluation (by NIST)
TNO	Nederlands Instituut voor Toegepaste Geowetenschappen
UEM	Unpartitioned Evaluation Map
VAD	Voice Activity Detection
VT	Vermont (USA)
VQ	Vector Quantization
#	Number

List of Appendices

A **HCopy configuration**

B **Detailed Results**

C **Contents of CD**

Appendix A

Configuration of Feature Extraction

For feature extraction I used *HCopy* tool from HTK. Example of usage:

$$./HCopy - C./hcopy\_mfcc19\_16.cfginput.rawoutput.fea$$

Contents of used configuration file:

```
SOURCEKIND   = WAVEFORM
SOURCEFORMAT = NOHEAD
#SOURCERATE  = 1250     # relates to sampling frequency of 8kHz
SOURCERATE   = 625      # relates to sampling frequency of 16kHz
BYTEORDER    = VAX
TARGETFORMAT = HTK
TARGETKIND   = MFCC
#TARGETKIND  = FBANK

LOFREQ       = 64       # low frequency cut-off in fbank analysis
#HIFREQ      = 4000     # high frequency cut-off in fbank analysis
HIFREQ       = 8000
NUMCHANS     = 23       # number of filter-bank channels
USEPOWER     = T        # using power spectrum
USEHAMMING   = T        # use hamming window on speech frame

PREEMCOEF    = 0        # no preemphasis
TARGETRATE   = 100000   # 10 ms frame rate
WINDOWSIZE   = 200000   # 20 ms window
SAVEWITHCRC  = F        # do not attach a checksum to output parameter file

#CEPLIFTER   = 22       # number of cepstral liftering coefficients
NUMCEPS      = 19       # number of cepstral coefficients
```

Appendix B

Detailed Results

This section contains scores of experiments displaying scores of complete dataset.

	Diarization error rate (%)		
File	**Original**	**Using ref. VAD**	**Using ref. Speaker Seg**
AMI_20041210-1052	28.53	30.26	24.09
AMI_20050204-1206	19.71	12.38	7.57
CMU_20050228-1615	27.60	30.21	23.09
CMU_20050301-1415	19.51	16.96	12.58
CMU_20050912-0900	40.92	22.34	18.38
CMU_20050914-0900	31.76	24.13	19.49
CMU_20061115-1030	53.54	31.06	20.41
CMU_20061115-1530	43.11	9.48	6.49
EDI_20050216-1051	42.79	55.42	37.90
EDI_20050218-0900	41.79	31.99	22.39
EDI_20061113-1500	57.54	43.22	48.35
EDI_20061114-1500	36.28	28.62	22.20
ICSI_20000807-1000	15.95	18.93	16.63
ICSI_20010208-1430	31.96	13.85	14.46
NIST_20030623-1409	22.10	10.44	6.28
NIST_20030925-1517	39.80	39.80	22.12
NIST_20051024-0930	23.53	19.22	23.01
NIST_20051102-1323	28.32	22.31	15.74
NIST_20051104-1515	31.55	10.76	5.89
NIST_20060216-1347	17.68	14.82	14.64
TNO_20041103-1130	42.45	34.54	30.83
VT_20050304-1300	33.42	22.30	20.70
VT_20050318-1430	33.24	21.51	21.60
VT_20050408-1500	11.82	3.14	1.73
VT_20050425-1000	28.25	29.27	23.98
VT_20050623-1400	37.98	30.81	21.34
VT_20051027-1400	65.13	33.10	47.48
Average	**33.79**	**24.09**	**20.08**

Table B.1: Scores of clustered segmentation of original system and system using reference VAD or reference speaker segmentation to reveal the most faulty subsystem (launched 16th November 2010 without final Viterbi re-segmentation)

		Config. 1		Config. 2	
File	**Ref. # Spk**	**DER (%)**	**# Spk**	**DER (%)**	**# Spk**
AMI_20041210-1052	4	26.73	5	7.34	3
AMI_20050204-1206	4	11.74	6	7.79	6
CMU_20050228-1615	4	14.55	4	24.80	3
CMU_20050301-1415	4	13.84	3	12.61	3
CMU_20050912-0900	4	25.21	3	43.24	2
CMU_20050914-0900	4	24.68	3	30.43	3
CMU_20061115-1030	4	34.02	5	29.89	4
CMU_20061115-1530	4	18.09	5	12.50	4
EDI_20050216-1051	4	31.86	5	15.92	5
EDI_20050218-0900	4	23.05	6	19.48	5
EDI_20061113-1500	4	43.50	7	36.81	5
EDI_20061114-1500	4	28.09	7	18.98	6
ICSI_20000807-1000	6	33.21	10	11.26	7
ICSI_20010208-1430	6	17.85	6	7.52	6
NIST_20030623-1409	5	4.62	7	2.15	6
NIST_20030925-1517	4	54.81	7	36.97	5
NIST_20051024-0930	9	18.46	9	19.41	7
NIST_20051102-1323	8	15.32	10	8.31	9
NIST_20051104-1515	4	24.81	8	4.92	4
NIST_20060216-1347	6	15.68	6	10.20	5
TNO_20041103-1130	4	41.91	8	28.62	6
VT_20050304-1300	5	20.70	6	1.10	5
VT_20050318-1430	5	28.08	5	10.78	4
VT_20050408-1500	5	4.60	5	4.16	5
VT_20050425-1000	4	17.90	3	26.90	2
VT_20050623-1400	5	28.71	5	22.34	4
VT_20051027-1400	4	50.38	7	42.11	4
Average	4.74	**24.41**	5.96	**18.52**	4.74

Table B.2: Comparison of scores of diarization system using different configurations. The first configuration (presented in section 5.1) was launched 28th December 2010 The second configuration (presented in section 5.4) was launched 26th January 2011. long list of differences between *Config. 1* and *Config. 2* is presented in section 5.4. Shortly, the main change is in clustering lambda which was extended from 4.8 to 11.

	Error rates (%)			
File	**MISS**	**FA**	**SPK**	**Total**
AMI_20041210-1052	0.4	0.9	6.1	7.34
AMI_20050204-1206	3.2	0.7	3.9	7.79
CMU_20050228-1615	12.7	0.3	11.7	24.80
CMU_20050301-1415	4.1	0.5	8.0	12.61
CMU_20050912-0900	21.1	4.2	17.9	43.24
CMU_20050914-0900	18.3	4.1	8.0	30.43
CMU_20061115-1030	18.0	5.4	6.5	29.89
CMU_20061115-1530	5.5	3.6	3.4	12.50
EDI_20050216-1051	3.0	4.0	8.9	15.92
EDI_20050218-0900	3.4	3.1	12.9	19.48
EDI_20061113-1500	9.1	0.6	27.2	36.81
EDI_20061114-1500	2.5	2.5	13.9	18.98
ICSI_20000807-1000	4.7	0.3	6.3	11.26
ICSI_20010208-1430	3.8	0.8	2.9	7.52
NIST_20030623-1409	1.0	0.8	0.3	2.15
NIST_20030925-1517	7.6	3.9	25.5	36.97
NIST_20051024-0930	4.2	0.4	14.7	19.41
NIST_20051102-1323	3.2	1.9	3.2	8.31
NIST_20051104-1515	3.4	0.6	1.0	4.92
NIST_20060216-1347	2.5	1.5	6.2	10.20
TNO_20041103-1130	5.6	1.4	21.6	28.62
VT_20050304-1300	0.4	0.5	0.2	1.10
VT_20050318-1430	2.3	0.8	7.7	10.78
VT_20050408-1500	3.1	0.4	0.7	4.16
VT_20050425-1000	5.3	0.4	21.2	26.90
VT_20050623-1400	5.4	2.6	14.3	22.34
VT_20051027-1400	3.1	3.4	35.6	42.11
Average	**6.0**	**1.9**	**10.6**	**18.52**

Table B.3: Detailed DER scores of my system using the best configuration (missed speaker time relative, false alarm time relative, speaker error rime relative and total diarization error rate; launched 26th January 2011)

	VAD error rate (%)	
File	**My system**	**Huijbregts' system**
AMI_20041210-1052	0.90	1.06
AMI_20050204-1206	2.33	1.73
CMU_20050228-1615	8.56	2.98
CMU_20050301-1415	3.87	1.85
CMU_20050912-0900	16.31	5.26
CMU_20050914-0900	15.74	5.67
CMU_20061115-1030	15.47	7.05
CMU_20061115-1530	6.13	4.53
EDI_20050216-1051	3.79	2.06
EDI_20050218-0900	3.44	3.19
EDI_20061113-1500	3.90	2.16
EDI_20061114-1500	2.67	4.02
ICSI_20000807-1000	1.72	5.91
ICSI_20010208-1430	3.46	2.72
NIST_20030623-1409	1.25	1.15
NIST_20030925-1517	4.81	2.95
NIST_20051024-0930	1.80	3.89
NIST_20051102-1323	1.98	2.30
NIST_20051104-1515	0.59	0.57
NIST_20060216-1347	3.10	2.49
TNO_20041103-1130	5.81	3.26
VT_20050304-1300	0.74	1.97
VT_20050318-1430	4.94	5.27
VT_20050408-1500	3.40	2.07
VT_20050425-1000	1.59	1.42
VT_20050623-1400	3.65	4.10
VT_20051027-1400	5.56	5.88
Average	**4.98**	**3.27**

Table B.4: Comparison of VAD scores of implemented diarization system using the best configuration (launched 26th January 2011) and system implemented by Marijn Huijbregts (launched 19th February 2009)

		My system		Huijbregts' system	
File	**Ref. # Spk**	**DER (%)**	**# Spk**	**DER (%)**	**# Spk**
AMI_20041210-1052	4	7.34	3	7.44	3
AMI_20050204-1206	4	7.79	6	3.88	4
CMU_20050228-1615	4	24.80	3	13.34	5
CMU_20050301-1415	4	12.61	3	4.70	4
CMU_20050912-0900	4	43.24	2	22.86	7
CMU_20050914-0900	4	30.43	3	28.92	7
CMU_20061115-1030	4	29.89	4	23.49	6
CMU_20061115-1530	4	12.50	4	10.99	6
EDI_20050216-1051	4	15.92	5	15.59	3
EDI_20050218-0900	4	19.48	5	7.19	4
EDI_20061113-1500	4	36.81	5	11.38	4
EDI_20061114-1500	4	18.98	6	8.44	4
ICSI_20000807-1000	6	11.26	7	18.56	3
ICSI_20010208-1430	6	7.52	6	14.21	3
NIST_20030623-1409	5	2.15	6	1.73	5
NIST_20030925-1517	4	36.97	5	12.64	3
NIST_20051024-0930	9	19.41	7	6.86	6
NIST_20051102-1323	8	8.31	9	7.88	6
NIST_20051104-1515	4	4.92	4	4.12	4
NIST_20060216-1347	6	10.20	5	4.77	7
TNO_20041103-1130	4	28.62	6	39.71	4
VT_20050304-1300	5	1.10	5	8.02	6
VT_20050318-1430	5	10.78	4	28.08	4
VT_20050408-1500	5	4.16	5	6.06	7
VT_20050425-1000	4	26.90	2	6.08	4
VT_20050623-1400	5	22.34	4	22.67	5
VT_20051027-1400	4	42.11	4	21.15	5
Average	4.74	**18.52**	4.74	**12.91**	4.78

Table B.5: Comparison of scores of my diarization system using best configuration (launched 26th January 2011) and system implemented by Marijn Huijbregts using his best configuration (launched 25th February 2009)

	DER (%)	
File	**My system**	**System of Chicote et al.**
CMU_20061115-1030	29.89	24.02
CMU_20061115-1530	12.50	10.87
EDI_20061113-1500	36.81	13.06
EDI_20061114-1500	18.98	29.29
NIST_20051104-1515	4.92	5.36
NIST_20060216-1347	10.20	7.75
VT_20050408-1500	4.16	3.95
VT_20050425-1000	26.90	16.05
Average	**17.75**	**13.61**

Table B.6: Comparison of scores of my system (launched 26th January 2011) and system implemented by Barra-Chicote et al. [4, page 10, table V] (using mfcc, tdoa+icc) on RT07 meeting set

	DER (%)	
File	**My system**	**System of Luque et al.**
CMU_20050912-0900	43.24	27.60
CMU_20050914-0900	30.43	25.56
CMU_20061115-1030	29.89	17.51
CMU_20061115-1530	12.50	8.06
EDI_20050216-1051	15.92	53.09
EDI_20050218-0900	19.48	21.99
EDI_20061113-1500	36.81	18.70
EDI_20061114-1500	18.98	7.78
NIST_20051024-0930	19.41	34.44
NIST_20051102-1323	8.31	29.53
NIST_20051104-1515	4.92	6.19
NIST_20060216-1347	10.20	9.04
VT_20050408-1500	4.16	8.76
VT_20050425-1000	26.90	9.34
VT_20050623-1400	22.34	42.05
VT_20051027-1400	42.11	36.03
Average	**20.80**	**23.01**

Table B.7: Comparison of scores of my system (launched 26th January 2011) and system implemented by Luque et al. [14, slide 16] using RT06-07 meeting sets

	DER (%)	
File	**My system**	**LIA-EURECOM RT'09 system**
AMI_20041210-1052	7.34	0.6
AMI_20050204-1206	7.79	8.1
CMU_20050228-1615	24.80	7.5
CMU_20050301-1415	12.61	7.8
CMU_20050912-0900	43.24	18.4
CMU_20050914-0900	30.43	14.2
EDI_20050216-1051	15.92	27.2
EDI_20050218-0900	19.48	10.2
ICSI_20000807-1000	11.26	23.5
ICSI_20010208-1430	7.52	32.7
ICSI_20010531-1030	–	15.6
ICSI_20011113-1100	–	21.3
LDC_20011116-1400	–	5.4
LDC_20011116-1500	–	11.3
NIST_20030623-1409	2.15	5.5
NIST_20030925-1517	36.97	26.0
NIST_20050427-0939	–	3.6
NIST_20051024-0930	19.41	8.3
NIST_20051102-1323	8.31	7.0
VT_20050304-1300	1.10	5.6
VT_20050318-1430	10.78	41.0
VT_20050623-1400	22.34	32.5
VT_20051027-1400	42.11	14.4
Average	**18.39**	**14.9**

Table B.8: Comparison of scores of my system (launched 26th January 2011) and LIA-EURECOM RT'09 system implemented by Fredouille et al. [10] which ran on selection of RT meetings

Appendix C

Contents of CD

Enclosed compact disc contains:

- Complete sources of system for speaker diarization implemented in Matlab
- Short description how to use the diarization system
- Final segmentation and results of test set of files processed by implemented speaker diarization system
- Code documentation of implemented system
- PDF version of this thesis
- TEXsource code of this thesis